



UNIVERSITAT DE  
BARCELONA

Treball final de grau

GRAU D'ENGINYERIA INFORMÀTICA

Facultat de Matemàtiques  
Universitat de Barcelona

---

**EATLOG: UN ASISTENTE DIGITAL  
PERSONAL PARA ASISTIR A  
MEJORAR LOS HÁBITOS  
SALUDABLES**

---

**Autor: Juan Luis Soriano Oliú**

**Supervisor: Dra. Petia I. Radeva**  
**Completado en: Departament de Matemàtica Aplicada i Anàlisi**

**Barcelona, 22 Junio 2017**

## Resumen

En la actualidad, los hábitos saludables relacionados con las dietas están cobrando cada vez más protagonismo. La sociedad está tomando conciencia de lo importante que es llevar una dieta equilibrada para muchos aspectos de su vida. El registro de las comidas es una de las grandes dificultades que sufren muchas de las personas que quieren llevar un seguimiento de su dieta.

EatLog se desarrolla con el objetivo de generar automáticamente el diario de comidas a través de una aplicación para móviles gracias al reconocimiento automático de imágenes. Se implementan algoritmos de reconocimiento basados en la tecnología *Deep Learning*, concretamente en las redes neuronales convolucionales. Esta tecnología ha permitido desarrollar algoritmos que permiten varios reconocimientos de una imagen, entre los que destaca la comida y la categoría. La aplicación crea entonces de forma automática un registro de las comidas conectando con un servidor que contiene los algoritmos de reconocimiento. Se han descargado recetas e información nutricional de ingredientes. De este modo la aplicación genera la información nutricional de las comidas reconocidas. El usuario puede entonces consultar información y establecerse objetivos con respecto a cualquiera de los 26 indicadores nutricionales que se gestionan.

Finalmente, para mejorar los resultados de reconocimiento, se ha procedido a descargar imágenes de comida en alta resolución. Esto ha permitido crear una base de datos de 200 categorías de comida que reconoce EatLog (101 correspondientes a la base de datos existente Food101 [9]). Las 99 restantes han sido descargadas en el proyecto, con una media de 800 imágenes por categoría.

## Agradecimientos

En primer lugar, desearía mostrar mi gratitud hacia todas las personas de la universidad que me han acompañado en este proyecto. Gracias a Dra. Petia I. Radeva, durante este año hemos compartido muchas reuniones y experiencias, de las que he aprendido mucho. Gracias a mi amigo y compañero Pedro Herruzo, entré en este proyecto atraído por tu trabajo realizado el anterior año y ha acabado siendo una gran aventura. Gracias a mis compañeros, estudiantes de Doctorado, Marc Bolaños y Eduardo Aguilar, vuestro apoyo y ayuda han sido constantes a lo largo de estos meses.

Finalmente quería mencionar y agradecer a todos los que han estado a mi lado estos meses, mis amigos y mi familia: mis hermanos Titín y Georgina y mis padres Jorge y Pilar. He sentido en todo momento vuestro apoyo incondicional, compartiendo los buenos momentos y animando en los no tan buenos.

# Índice

<b>Resumen</b>	<b>1</b>
<b>Agradecimientos</b>	<b>2</b>
<b>Índice de figuras</b>	<b>5</b>
<b>Índice de tablas</b>	<b>6</b>
<b>1. Introducción</b>	<b>7</b>
1.1. Contexto . . . . .	7
1.2. Estado del Arte . . . . .	8
1.3. Objetivos . . . . .	9
<b>2. Análisis y Diseño</b>	<b>12</b>
2.1. Análisis del sistema . . . . .	12
2.1.1. Requisitos . . . . .	12
2.1.2. Casos de uso . . . . .	15
2.2. Diseño del Sistema . . . . .	21
2.2.1. Arquitectura física del sistema . . . . .	21
2.2.2. Diagrama de Clases de la aplicación . . . . .	22
2.2.3. Bases de Datos de la aplicación y el servidor . . . . .	24
<b>3. Herramientas e Implementación</b>	<b>28</b>
3.1. Reconocimiento de comida a través de <i>Deep Learning</i> . . . . .	28
3.1.1. Tecnología de reconocimiento, redes neuronales convolucionales . . . . .	28
3.1.2. Resultados . . . . .	30
3.2. Aplicación . . . . .	30
3.3. Servidor . . . . .	32
3.4. Construcción de los DataSets . . . . .	35
3.4.1. Descarga de Imágenes de Comida . . . . .	35
3.4.1.1. Descarga de imágenes procedentes de Google . . . . .	36
3.4.1.2. Descarga de imágenes procedentes de Instagram . . . . .	38
3.4.1.3. Curación de las imágenes usando la herramienta web de etiquetado . . . . .	39
3.4.1.4. Resultados . . . . .	40
3.4.2. Construcción de los Datasets de Recetas e Ingredientes . . . . .	41
3.4.2.1. Información a descargar . . . . .	41
3.4.2.2. Implementación . . . . .	41
3.4.2.3. Resultados . . . . .	44
<b>4. Manual de usuario</b>	<b>45</b>
<b>5. Conclusiones y trabajos futuros</b>	<b>55</b>
5.1. Conclusiones . . . . .	55
5.2. Trabajos futuros . . . . .	55



6. Bibliografía	57
Apéndice	58

## Índice de figuras

1.	Ejemplos de FatSecret y LoseIt!	9
2.	Diagrama Lógico de EatLog	11
3.	Esquema casos de usos Usuario	15
4.	Esquema casos de usos Servidor	16
5.	Arquitectura física del sistema	21
6.	Diagrama de Clases de la aplicación	23
7.	Diagrama Entidad Relación de la Base de Datos de la Aplicación	24
8.	Diagrama Entidad Relación de la Base de datos del servidor	26
9.	Modelo biológico de una red neuronal	28
10.	Modelo de un perceptrón	29
11.	Reparto Mercado Android por Versiones [5]	31
12.	Cuota de mercado de Android	32
13.	Envío de información sobre el usuario	33
14.	Envío de la imagen al servidor	33
15.	Envío de resultados desde el servidor	34
16.	Envío de confirmación desde la App	35
17.	Diagrama de Flujo del código correspondiente a la descarga de imágenes de Google	37
18.	Capturas de imagen de Google	38
19.	Diagrama de Flujo del código de descarga de imágenes de Instagram	39
20.	Captura de la Web utilizada para el etiquetado de imágenes	40
21.	Diagrama de flujo del código de descarga de recetas e información nutricional de ingredientes	43
22.	Ejemplo de la información nutricional almacenada para cada ingrediente.	44
23.	Ejemplo de la información almacenada de cada receta.	44
24.	Pantallas de la aplicación: pantalla presentación y pantalla principal	45
25.	Pantalla de la aplicación: pantalla de cámara	46
26.	Pantallas de la aplicación: pantalla de entrada manual de comida y pantalla de escritura manual de comida	46
27.	Pantalla de la aplicación: pantalla de resultados de reconocimiento de la imagen	47
28.	Pantallas de la aplicación: pantalla de elección de hora y pantalla de elección de fecha	48
29.	Pantallas de la aplicación: pantallas de la información nutricional y del la receta de la comida	49
30.	Pantalla de la aplicación: pantalla de añadir ingrediente	50
31.	Pantallas de la aplicación: pantallas de estadísticas diarias organizadas por indicador nutricional	50
32.	Pantallas de la aplicación: pantalla del diario semanal y pantalla de detalle de una de las comidas	51
33.	Pantallas de la aplicación: pantalla principal de objetivos y pantalla de añadir objetivo	52
34.	Pantalla de la aplicación: pantalla de información de las comidas que reconoce EatLog	53
35.	Pantalla de la aplicación: pantalla de álbum de imágenes tomadas a través de EatLog	54

## Índice de tablas

1.	Requisitos Funcionales . . . . .	13
2.	Requisitos no Funcionales de Usabilidad . . . . .	14
3.	Otros Requisitos no Funcionales . . . . .	14
5.	Caso de Uso 5. Consultar y confirmar reconocimiento imagen . . . . .	17
7.	Caso de Uso 6. Confirmar información receta . . . . .	19
9.	Caso de Uso 13. Procesar Reconocimiento Imagen . . . . .	20
10.	Datos sobre la cantidad de imágenes antes y después del filtrado y etiquetado . . . . .	40
12.	Caso de Uso 1. Registrarse . . . . .	59
14.	Caso de Uso 2. Hacer Foto . . . . .	60
16.	Caso de Uso 3. Subir Foto . . . . .	61
18.	Caso de Uso 4. Subir Comida . . . . .	62
20.	Caso de Uso 7. Consultar Estadísticas . . . . .	63
22.	Caso de Uso 8. Consultar Comidas . . . . .	65
24.	Caso de Uso 9. Borrar Comida . . . . .	66
26.	Caso de Uso 10. Consultar Platos Reconocibles . . . . .	67
28.	Caso de Uso 11. Proponer Nuevo Plato . . . . .	68
30.	Caso de Uso 12. Añadir Objetivos . . . . .	70
32.	Caso de Uso 14. Guardar Información sobre Imagen . . . . .	71
34.	Caso de Uso 15. Guardar Información sobre usuario . . . . .	72

# 1. Introducción

## 1.1. Contexto

En la época y contexto actuales la alimentación está cobrando cada vez más protagonismo en el ámbito de la salud. Los continuos estudios médicos sobre nutrición están concienciando a la sociedad, permitiendo entender hasta qué punto es importante lo que ingerimos y la repercusión que tiene esto en nuestro organismo. De acuerdo a los datos proporcionados por la plataforma .<sup>E</sup>uropean EU Platform on Diet, Physical Activity and Health”<sup>1</sup>, en 2013, 4.2 millones de enfermedades crónicas en la unión Europea, como la diabetes o el cáncer, están en relación directa con la falta de ejercicio físico o una dieta poco saludable. Tal y como muestra el artículo [1], 50 % de las causas de todas las enfermedades crónicas se basan en los hábitos saludables, dejando el resto de causas entre: 20 % al entorno, 20 % a la genética y 10 % al acceso a la asistencia médica. Uno de los principales problemas de los programas de intervención, es el mantenimiento de un cambio de comportamiento a largo plazo [2]. Vivimos en un entorno propicio al sobrepeso y la obesidad con ocupaciones cada vez más sedentarias, que combinadas hacen que sea difícil mantener las metas de control de peso a largo plazo. Una simple revisión de ingesta diaria de alimentos es un importante primer paso para la pérdida de peso [3].

Es evidente pensar en el sector de la nutrición cuando hablamos de hábitos alimenticios. Los nutricionistas y dietistas estudian el proceso y la importancia de lo que ingerimos. El servicio que nos presta este sector es de asesoramiento, no es de curación propiamente dicha. Es responsabilidad de cada uno de nosotros seguir los consejos que nos da un nutricionista o dietista. Este aspecto es el que dificulta en cierto modo la efectividad de los servicios dados en este sector. Se puede llevar un seguimiento mental o escrito de lo que se ingiere, con el fin de estudiarlo y juzgarlo a posteriori con un profesional. En este punto nos encontramos con una dificultad, la creación del diario de comidas. Este concepto se refiere al registro de las comidas que se llevan a cabo, especificando la hora y día. Los principales métodos requieren de una dedicación personal estricta y responsable. Estos tipos de anotaciones tienden a ser subjetivos, escasos, discontinuos, con una retrospectiva sesgada, y no se acaban de cumplir. El principal inconveniente de la anotación manual es el hecho de la notificación incompleta de los alimentos consumidos. La sociedad está limitada en su capacidad de estimar con precisión el tamaño de las porciones. Esto sigue suponiendo un problema en la actualidad, siendo la confianza en el paciente la principal fuente de veracidad de la creación del llamado diario de comidas.

---

<sup>1</sup>[https://ec.europa.eu/health/nutrition\\_physical\\_activity/platform.en](https://ec.europa.eu/health/nutrition_physical_activity/platform.en)

## 1.2. Estado del Arte

Entre las principales razones de una vida saludable encontramos los aspectos de tener una vida activa con una cantidad razonable de actividades físicas y seguir una dieta sana. Actualmente existen en el mercado múltiples aplicaciones enfocadas en el ámbito de la salud con el objetivo de conseguir un estilo de vida saludable. La mayoría de estas aplicaciones se centran en los aspectos ya citados, las actividades físicas y los hábitos nutricionales:

- Ejercicios físicos: Aplicaciones como Runtastic, RunKeeper, Pacer (Pedometer), Daily Yoga, etc. ofrecen una forma de llevar un control de las actividades físicas realizadas por el usuario. Este control puede ser en algunos casos automático usando seguidores físicos (trackers) como FitBit, Jawbone, etc., los cuales hacen uso de sensores GPS o acelerómetros para medir la actividad física; o manual, requiriendo que el usuario introduzca manualmente la actividad realizada. Adicionalmente, la mayoría de estas aplicaciones permiten visualizar y controlar el historial de la actividad realizada. Se puede considerar que la mayoría de las aplicaciones de este sector cubren todas las necesidades.

- Nutrición: En este grupo podemos encontrar aplicaciones como MyFitnessPal <sup>2</sup>, Cronometer <sup>3</sup>, Fatsecret <sup>4</sup>, Lose It! <sup>5</sup> o Calorie Mama <sup>6</sup>, etc., todas ellas con el objetivo de permitir al usuario perder peso. La mayoría, requieren que el usuario introduzca manualmente la comida ingerida y que corrija la información nutricional asociada. Como alternativa de la anotación manual, tres de las aplicaciones acaban de introducir la opción del reconocimiento automático de la comida ingerida (LoseIt!, Fatsecret y Calorie Mama). Sin embargo, estas aplicaciones no tienen un funcionamiento óptimo, se puede decir que su rendimiento es limitado, podemos ver un ejemplo en [Figura 1](#). El reconocimiento no es capaz de diferenciar las imágenes que no son comida, la interfaz es cargada y la participación manual del usuario sigue siendo alta y costosa. Por ejemplo, alguna de ellas reconoce correctamente una ensalada. No obstante, se muestra tras esto una larga lista de tipos de ensalada, donde el usuario debe elegir y especificar a continuación el tamaño. Esto sigue requiriendo una participación del usuario y tanta información hace que la interfaz sea menos agradable. Cabe destacar el crecimiento de este sector de aplicaciones, por ejemplo LoseIt! cuenta con millones de descargas (entre 5 y 10 de millones). Esto demuestra el aumento de la demanda por aplicaciones relacionadas con la nutrición. Sin embargo, no encontramos un funcionamiento óptimo en ninguna de ellas, que cubra todas las necesidades que puede tener un usuario en este contexto.

---

<sup>2</sup><https://www.myfitnesspal.com/es>

<sup>3</sup><https://cronometer.com/>

<sup>4</sup><https://www.fatsecret.es/>

<sup>5</sup><https://www.loseit.com/>

<sup>6</sup><http://www.caloriemama.ai/app>

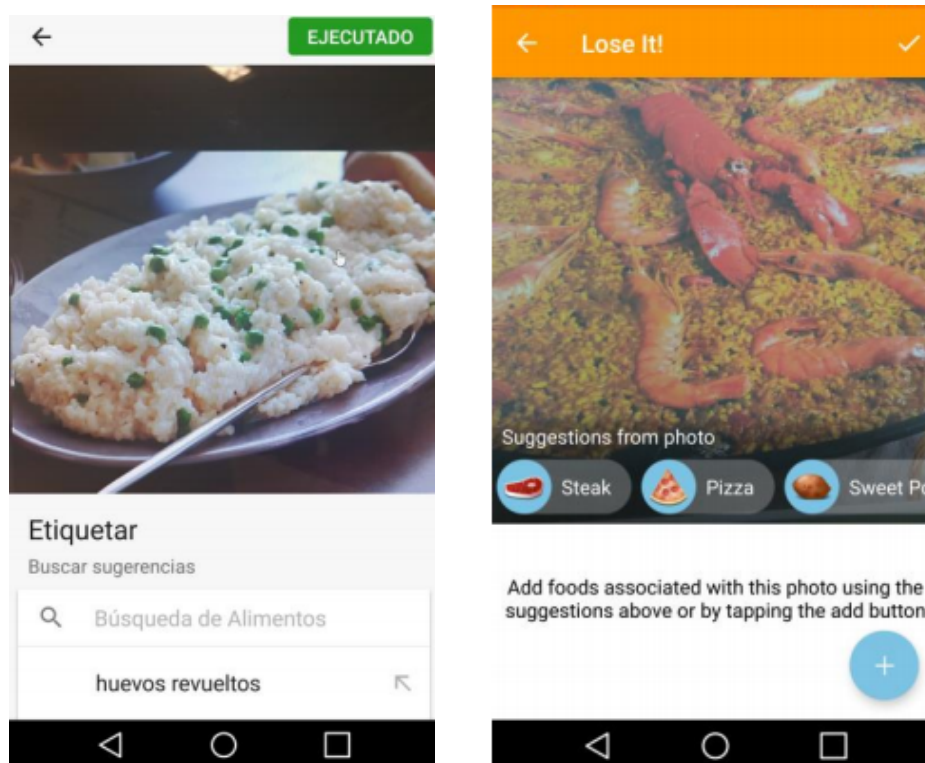


Figura 1: Ejemplos de FatSecret y LoseIt!

Se ha visto que existen varias aplicaciones enfocadas a la nutrición, varias de ellas con una gran cantidad de usuarios. La ventaja competitiva que ofrece EatLog es el aspecto simple e intuitivo de una interfaz más sencilla. Por defecto, se elegirá la receta más común del plato introducido, siempre dejando la opción de modificar los ingredientes. Esto permitirá un diseño y experiencia más simples. Por otro lado, el reconocimiento que ofrecerá la aplicación será más completo (se ejecutan otros reconocedores que permiten dictaminar si la imagen es comida o no, a qué categoría de comida pertenece, etc)

### 1.3. Objetivos

El objetivo principal es diseñar e implementar una aplicación móvil para construir automáticamente diarios de nutrición y obtener los hábitos alimentarios basada en el reconocimiento automático de comida, de sus ingredientes y de su composición nutricional a partir de una fotografía.

Las objetivos específicos de la aplicación EatLog se pueden dividir en diferentes secciones:

**- Reconocimiento de comida. Reconocer automáticamente la comida mediante el uso de técnicas de visión artificial.**

Para el reconocimiento automático, EatLog se basará en técnicas de Visión por Computador y Aprendizaje Automático. En particular, se aplicará una tecnología muy reciente llamada Aprendizaje Profundo (*Deep Learning*) que actualmente está revolucionando el avance científico en diferentes campos como la inteligencia artificial, la robótica, la visión por computador, la comprensión y traducción de lenguaje natural, etc. La aparición de esta tecnología ha permitido que la Visión por Computador sea capaz de llevar a cabo reconocimientos de imagen tan complejos. El usuario introduce la comida a través de una imagen y se aplica el reconocimiento usando las técnicas de Aprendizaje Profundo desarrolladas. EatLog cuenta con varios tipos de reconocimiento. Es capaz de reconocer:

- Si la imagen corresponde a comida
- La categoría principal de la imagen (entre 11 categorías generales)
- El tipo de plato de la imagen. Actualmente se reconocen 200 platos diferentes.

**- Traducción del plato reconocido a contenido nutricional.** EatLog permitirá al usuario tener un seguimiento detallado de los ingredientes incluidos en su comida y, en consecuencia, tener información nutricional detallada de cada uno de ellos. Si lo desea, el usuario podrá precisar los ingredientes incluidos en la comida ingerida una vez aplicado el reconocimiento automático. EatLog propondrá automáticamente la composición más habitual de ingredientes de la comida que ha sido reconocida. Posteriormente, se ofrecerá la posibilidad de modificar la cantidad de cualquiera de los ingredientes incluidos, añadir nuevos ingredientes, o eliminar ingredientes que no estén en la receta específica del usuario. A partir de esta información (ya sea generada completamente de forma automática o modificada por el usuario), se generará automáticamente la composición nutricional de la comida. De este modo, se registrará la composición nutricional precisa consumida (a través de los macronutrientes) y permitirá aplicar un seguimiento más minucioso. La base de datos de las recetas se obtendrá no en recetas anónimas, sino de fuentes científicas con recetas avaladas y consensuadas por grupos de expertos. Por otro lado, estas fuentes nos proporcionarán la composición nutricional de los ingredientes. Juntando estas dos bases de datos se conseguirá generar la información nutricional refinada de los platos.

**- Creación de un diario de comidas.** Tras aplicar el reconocimiento de la imagen y el refinamiento de la receta EatLog registra la comida y genera el diario de comidas del usuario. Se ofrecerán gráficos visuales en los que podrá consultar su avance y progreso. En ellos se mostrarán estadísticas sobre los diferentes tipos de indicadores nutricionales.

Se explica en la [Figura 2](#) el diagrama lógico del funcionamiento de EatLog con las principales funcionalidades que ha de cubrir, y que objetivos y tareas secundarias surgen de estas funcionalidades.

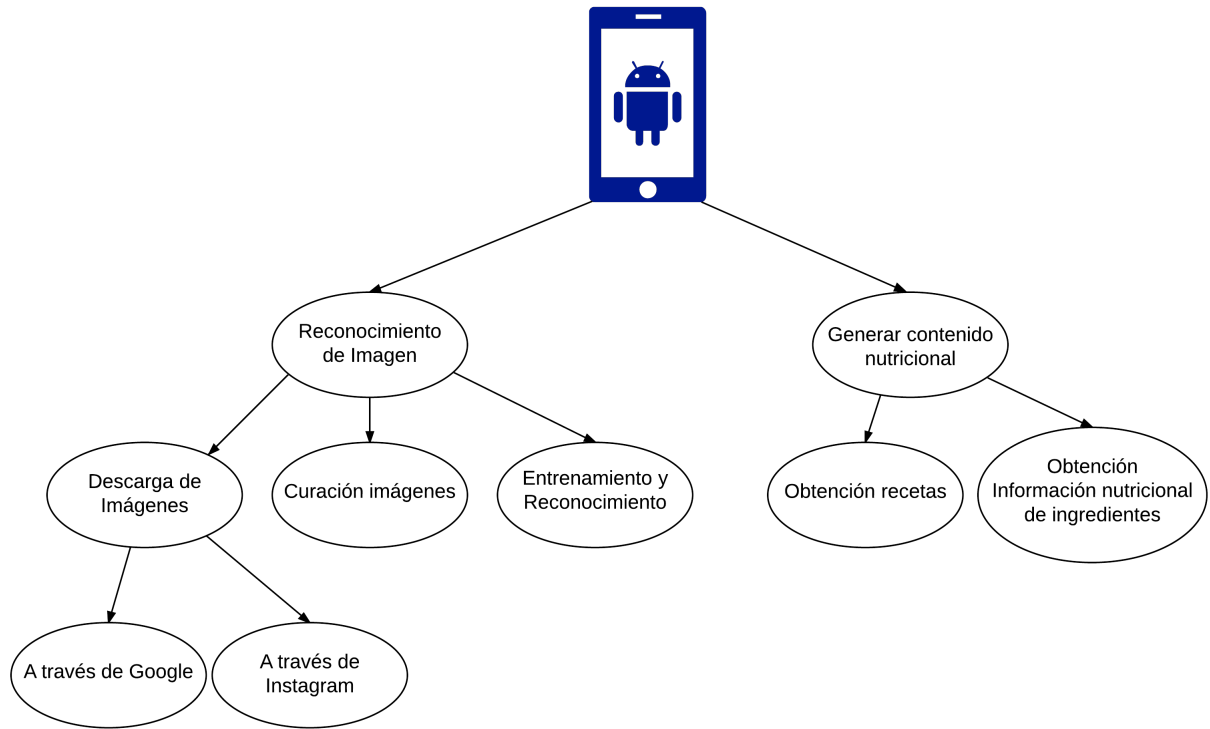


Figura 2: Diagrama Lógico de EatLog

Para proceder al reconocimiento de imagen, se deberán descargar imágenes. Contaremos con dos fuentes de descarga, *Google* y *Instagram*. Tras la descarga, se deberá tratar las imágenes para eliminar todas las que no son convenientes, es decir el proceso de curación de imágenes. Finalmente se llevará a cabo el entrenamiento de los algoritmos que permitirá tener un sistema de reconocimiento de imagen.

Por otro lado, la EatLog deberá ser capaz de generar el contenido nutricional a partir de una imagen. Para llevar esto a cabo, se obtendrán las recetas, es decir, el listado de ingredientes, y la información nutricional de los ingredientes.



## 2. Análisis y Diseño

En primer lugar se procederá a estudiar los principales requisitos que debe cumplir la aplicación, teniendo en cuenta los requisitos funcionales y no funcionales (enfocados a la usabilidad principalmente). Gracias a este estudio, se podrán analizar las acciones que debe cubrir el sistema por parte del usuario y del servidor, es decir, los casos de uso. Este análisis será llevado a cabo de forma precisa, captando el flujo esperado y los posibles flujos secundarios de cada caso de uso. Tras el análisis, se podrá exponer el diseño de la arquitectura física del sistema y, más en concreto, de la aplicación y las bases de datos.

### 2.1. Análisis del sistema

#### 2.1.1. Requisitos

En la [Tabla 1](#) se exponen los requisitos funcionales de EatLog, especificando la prioridad que tienen y su justificación. Varios de ellos surgen de la propia definición de la aplicación, y los objetivos que se proponen.

Una vez analizados los requisitos funcionales, se procede a enumerar los requisitos no funcionales en la [Tabla 2](#). En primer lugar, se han desarrollado los requisitos no funcionales de usabilidad. Se han tenido en cuenta las 10 heurísticas de Jakob Nielsen <sup>7</sup>, una de las personas más respetadas en el ámbito mundial sobre usabilidad en la web.

Por otro lado, se han tenido en cuenta requisitos funcionales de otros tipos en la [Tabla 3](#). El requisito RNF7 no es un requisito que corresponda al desarrollo de la aplicación, donde está principalmente enmarcado EatLog. Este requisito supone una ampliación de posibilidades, pudiendo externalizar el servicio de reconocimiento de imágenes a terceras partes interesadas, como pueden ser grupos de nutricionistas con su propia aplicación móvil.

---

<sup>7</sup><http://www.braintive.com/10-reglas-heurísticas-de-usabilidad-de-jakob-nielsen/>

Descripción	Prioridad	Justificación
<b>RF1.</b> La aplicación debe permitir realizar una foto.	Obligatorio	La propia definición de la aplicación.
<b>RF2.</b> La aplicación debe permitir al usuario elegir entre los reconocimientos de la imagen propuestos.	Obligatorio	El sistema no es capaz de reconocer el plato de la imagen con una muy alta probabilidad. El sistema propone las opciones más probables y requiere la interacción del usuario.
<b>RF3.</b> La aplicación debe permitir al usuario gestionar las recetas propuestas de las comidas subidas	Deseable	Uno de los grandes objetivos de la aplicación es permitir al usuario la personalización de sus comidas.
<b>RF4.</b> La aplicación debe permitir al usuario subir una imagen contenida en su móvil.	Deseable	Permite una adaptabilidad de la aplicación. De este modo no requiere imperativamente la captura instantánea de la comida.
<b>RF5.</b> La aplicación debe permitir al usuario borrar una comida.	Deseable	Dota de personalización a la aplicación. El usuario se siente más cómodo.
<b>RF6.</b> La aplicación debe permitir al usuario consultar su diario semanal.	Deseable	Uno de los principales objetivos de la aplicación.
<b>RF7.</b> La aplicación debe permitir al usuario consultar información de cada indicador nutricional.	Deseable	Representa una información que puede ser valiosa e importante para muchos usuarios.
<b>RF8.</b> La aplicación debe permitir al usuario gestionar los objetivos sobre indicadores	Deseable	Permite al usuario llevar un seguimiento más personalizado y eficiente.
<b>RF9.</b> La aplicación debe permitir al usuario consultar información sobre las comidas que reconoce el sistema.	Deseable	La aplicación muestra los platos que el sistema reconoce, de modo que el usuario sea consciente.
<b>RF10.</b> El sistema debe gestionar la comunicación entre la aplicación y el servidor.	Obligatorio	La propia organización de la aplicación así lo requiere.
<b>RF11.</b> El sistema debe almacenar los datos aportados por el usuario ('feedback' del reconocimiento)	Obligatorio	El sistema debe ser capaz de mejorar los algoritmos de reconocimiento.
<b>RF12.</b> El sistema debe permitir al usuario identificarse	Obligatorio	Es necesario algún método para diferenciar los usuarios
<b>RF13.</b> El sistema debe procesar los algoritmos de reconocimiento de una imagen	Obligatorio	La definición del sistema así lo requiere.
<b>RF14.</b> El sistema debe almacenar datos del usuario y personalizar sus funcionalidades	Deseable	El sistema se adapta a los gustos y hábitos del usuario, permitiendo una mejor experiencia de usuario.
<b>RF15.</b> El sistema debe permitir al usuario registrar una comida sin asociar una imagen.	Deseable	Aporta adaptabilidad al sistema y permite al usuario mantener una privacidad si lo desea.

Tabla 1: Requisitos Funcionales

Descripción	Prioridad	Justificación
<b>RNF1.</b> La aplicación debe mostrar en qué sección se encuentra el usuario.	Deseable	Basado en la primera heurística de Nielsen de usabilidad: "Visibilidad del estatus del sistema". El usuario es capaz de ver en cada momento en qué sección de la aplicación se encuentra.
<b>RNF2.</b> La aplicación debe mostrar un indicativo de que se está ejecutando el reconocimiento durante la comunicación con el servidor.	Deseable	Basado en primera heurística de Nielsen de usabilidad: "Visibilidad del estatus del sistema". El usuario debe recibir un feedback durante la espera. De otro modo, no será capaz de diferenciarlo con un mal rendimiento.
<b>RNF3.</b> La aplicación debe mantener una estructura común en todas sus pantallas.	Deseable	Basado en la cuarta heurística de Nielsen de usabilidad: "Consistencia y estándares".
<b>RNF4.</b> La aplicación debe usar iconos y figuras fácilmente reconocibles y asimilables.	Deseable	Basado en la sexta heurística de Nielsen de usabilidad: "Mejor reconocer que memorizar".

Tabla 2: Requisitos no Funcionales de Usabilidad

Descripción	Prioridad	Tipo
<b>RNF5.</b> La aplicación debe ser apta a anteriores versiones de Android	Deseable	Rendimiento
<b>RNF6.</b> Las comunicaciones entre la aplicación y el servidor no deben durar más de 5 segundos.	Deseable	Eficiencia
<b>RNF7.</b> El sistema debe ser capaz de ofrecer los servicios de reconocimiento de imágenes sin la necesidad de la interfaz de la aplicación.	Deseable	Portabilidad

Tabla 3: Otros Requisitos no Funcionales

### 2.1.2. Casos de uso

Tras el análisis realizado sobre los requisitos de la aplicación y del sistema en general, se pueden concebir las funcionalidades que se deben cumplir. Estas funcionalidades se traducen en las acciones que puede llevar a cabo el usuario por un lado (Figura 3), y el servidor por otro lado (Figura 4). Algunas acciones extienden de otras, es decir, pueden ampliar la acción inicial llevada a cabo. Por ejemplo, el usuario puede consultar las comidas que ha registrado, y ampliar esta acción borrando una comida. Por otro lado, algunas acciones incluyen automáticamente la realización de otras acciones. Podemos ver que la acción de subir una comida (registrar una comida sin imagen) implica la acción de confirmar (o modificar) la información de la receta propuesta.

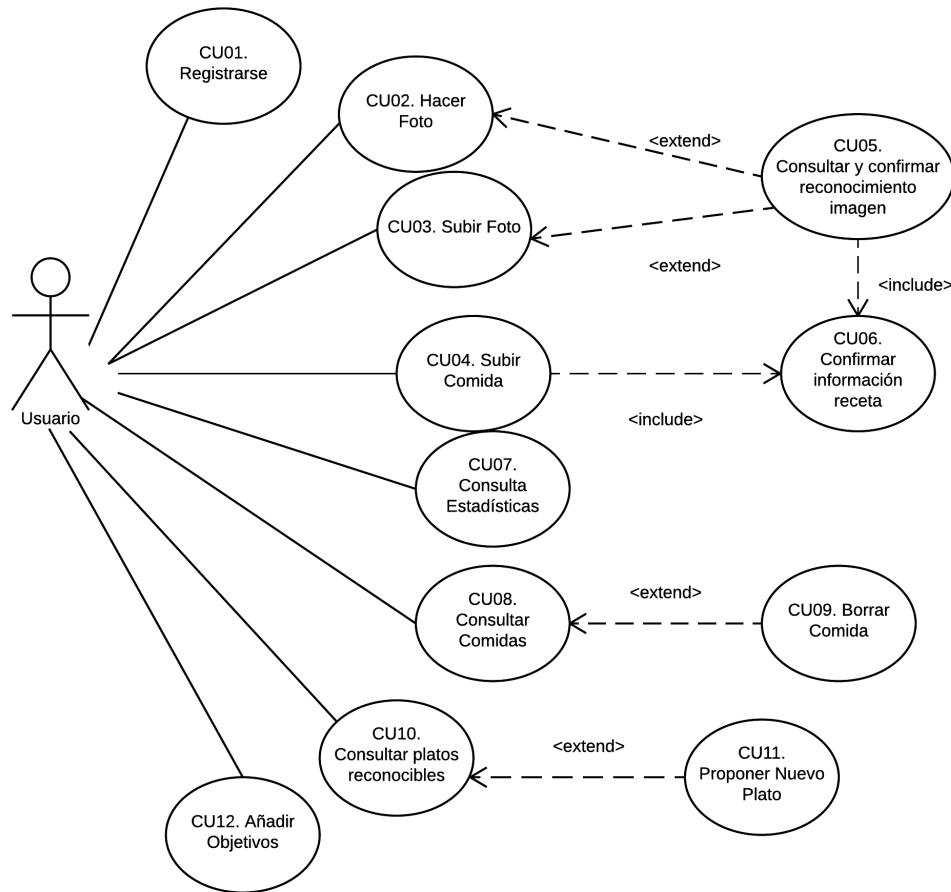


Figura 3: Esquema casos de usos Usuario

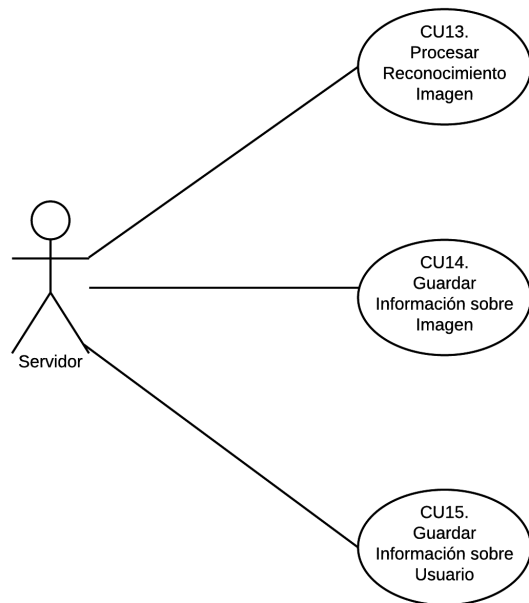


Figura 4: Esquema casos de usos Servidor

Los casos de usos estando especificados de forma genérica, se procederá ahora a analizar cada uno de ellos de manera detallada. Se mostrarán a continuación los detalles de las acciones que se han considerado más importantes y relevantes. Éstos son 'Consultar y confirmar reconocimiento image' (Tabla 5), 'Confirmar información recete' (Tabla 7) y 'Procesar Reconocimiento Imagen' (Tabla 9). El resto de casos de usos están detallados en el apéndice.

Caso de Uso 5	Consultar y confirmar reconocimiento imagen
<i>Actor Primario:</i>	Usuario
<i>Requisitos asociados:</i>	<b>RF2</b> La aplicación debe permitir al usuario elegir entre los reconocimientos de la imagen propuestos.
<i>Descripción:</i>	Una vez el usuario ha realizado o subido una foto, el sistema procesa los reconocimientos de la imagen, y expone al usuario los platos y categorías más probables. El usuario elige y confirma entre dichas opciones.
<i>Precondiciones:</i>	<ul style="list-style-type: none"> <li>El usuario ha realizado o subido una foto a través de EatLog</li> </ul>

---

**Secuencia normal:**

1. El sistema procesa en el servidor los algoritmos de reconocimiento
2. El sistema gestiona la comunicación de los resultados a la aplicación
3. El sistema muestra una pantalla los resultados del reconocimiento (5 platos propuestos y 1 categoría propuesta)
4. El usuario elige uno de los platos propuestos y confirma la categoría propuesta
5. El usuario confirma los reconocimientos ofrecidos por el sistema.

---

**Postcondiciones:**

---

**Excepciones/Extensiones:**

- 4.a Ninguno de los 5 platos propuestos coincide con la imagen del usuario:
  1. El usuario elige la opción otras comidas 'Other'
  2. El sistema muestra un campo de texto y una lista de comidas
  3. El usuario elige o escribe la comida
  4. El usuario confirma la opción escogida
- 4.b La categoría propuesta no coincide con la imagen del usuario
  1. El usuario clicla sobre el desplegable de categorías
  2. El sistema muestra el resto de categorías, ordenadas por probabilidad en el reconocimiento
  3. El usuario elige la categoría

---

<b>Frecuencia:</b>	Alta
--------------------	------

---

<b>Importancia:</b>	Alta
---------------------	------

---

<b>Comentarios:</b>
---------------------

---

Tabla 5: Caso de Uso 5. Consultar y confirmar reconocimiento imagen

---

<b>Caso de Uso 6</b>	<b>Confirmar información receta</b>
----------------------	-------------------------------------

---

<b>Actor Primario:</b>	Usuario
------------------------	---------

---

<b><i>Requisitos asociados:</i></b>	<b>RF3</b> La aplicación debe permitir al usuario gestionar las recetas propuestas de las comidas subidas.
<b><i>Descripción:</i></b>	El usuario modifica la receta propuesta, es decir, edita la lista de ingredientes.
<b><i>Precondiciones:</i></b>	<ul style="list-style-type: none"> <li>■ El usuario ha confirmado la información de los reconocimientos de la imagen capturada/subida o ha subido una comida sin imagen</li> </ul>
<b><i>Secuencia normal:</i></b>	
<ol style="list-style-type: none"> <li>1. El sistema procesa la elección del plato por parte del usuario y genera una receta</li> <li>2. El sistema muestra una pantalla con la imagen, la receta propuesta (listado de ingredientes) y la información nutricional correspondiente.</li> <li>3. El usuario confirma la receta.</li> <li>4. Se envía al servidor la información acerca del plato y la receta asociados a la imagen.</li> <li>5. El sistema registra la comida en EatLog.</li> <li>6. El sistema actualiza los valores nutricionales del día.</li> </ol>	
<b><i>Postcondiciones:</i></b>	<ul style="list-style-type: none"> <li>■ La comida está registrada en la aplicación, con la imagen y receta asociadas.</li> <li>■ La comida y su información, asociados con el usuario están registradas en el servidor.</li> <li>■ La imagen, con la predicción y confirmación del usuario están registradas en el servidor.</li> </ul>
<b><i>Excepciones/Extensiones:</i></b>	

- 3.a El usuario no está de acuerdo con la receta propuesta (Falta un ingrediente)
1. El usuario elige la opción añadir ingrediente (señalizada con símbolo '+')
  2. El sistema muestra una ventana emergente con dos campos de texto (nombre de ingrediente y peso)
  3. El usuario escribe la información del ingrediente y peso
  4. El usuario confirma el ingrediente
  5. El sistema añade el ingrediente a la lista de ingredientes
  6. El sistema actualiza la información nutricional mostrada
  7. El sistema almacena la modificación de la receta realizada
- 3.b El usuario no está de acuerdo con la receta propuesta (Sobra un ingrediente)
1. El usuario clicla sobre el icono de borrar (representado por una papelera) del ingrediente que desea eliminar
  2. El sistema actualiza la lista de ingredientes
  3. El sistema actualiza la información nutricional mostrada
  4. El sistema almacena la modificación de la receta realizada

<b><i>Frecuencia:</i></b>	Alta
<b><i>Importancia:</i></b>	Alta
<b><i>Comentarios:</i></b>	Cuando el usuario desea añadir un nuevo ingrediente, se muestra un autocompletado con los ingredientes que EatLog tiene almacenados (alrededor de 900), ya que no se tiene información nutricional de cualquier otro ingrediente que no esté en la base de datos.

Tabla 7: Caso de Uso 6. Confirmar información receta

<b>Caso de Uso 13</b>	<b>Procesar Reconocimiento Imagen</b>
<b><i>Actor Primario:</i></b>	Servidor
<b><i>Requisitos asociados:</i></b>	<b>RF13</b> El sistema debe procesar los algoritmos de reconocimiento de imagen.
<b><i>Descripción:</i></b>	El servidor ejecuta los algoritmos de reconocimiento aplicados a una imagen.



<b>Precondiciones:</b>	<ul style="list-style-type: none"> <li>■ El servidor ha recibido una imagen</li> <li>■ El servidor ha recibido una petición de reconocimiento con un usuario asociado (por ejemplo, de la aplicación EatLog)</li> </ul>
<hr/>	
<b>Secuencia normal:</b>	<ol style="list-style-type: none"> <li>1. El servidor hace una petición a la base de datos para conocer los reconocimientos permitidos del usuario</li> <li>2. La base de datos devuelve la información al servidor</li> <li>3. El servidor hace una petición a los algoritmos de reconocimiento (enviando la imagen, y los reconocimientos a procesar)</li> <li>4. Los algoritmos de reconocimiento retornan la información al servidor</li> <li>5. El servidor envía la información recibida</li> </ol>
<hr/>	
<b>Postcondiciones:</b>	<ul style="list-style-type: none"> <li>■ El sistema que ha realizado la petición de reconocimiento ha recibido la información</li> </ul>
<hr/>	
<b>Excepciones/Extensiones:</b>	<ol style="list-style-type: none"> <li>1.a El usuario que ha realizado la petición no existe en la base de datos del servidor: <ol style="list-style-type: none"> <li>1. El servidor responde enviando información al respecto.</li> </ol> </li> <li>1.b El usuario que ha realizado la petición ha sobrepasado el límite de reconocimientos: <ol style="list-style-type: none"> <li>1. El servidor responde enviando información al respecto.</li> </ol> </li> </ol>
<hr/>	
<b>Frecuencia:</b>	Alta
<hr/>	
<b>Importancia:</b>	Alta
<hr/>	
<b>Comentarios:</b>	
<hr/>	

Tabla 9: Caso de Uso 13. Procesar Reconocimiento Imagen

## 2.2. Diseño del Sistema

Una vez analizados los requisitos y especificado como se cumplirán a través de los casos de uso del usuario de la aplicación y del servidor, se presentará el diseño que surge de ello. Primero de todo, se explicará la arquitectura física general de EatLog (Figura 5), que muestra claramente cuál es el flujo del sistema en cada momento de ejecución.

### 2.2.1. Arquitectura física del sistema

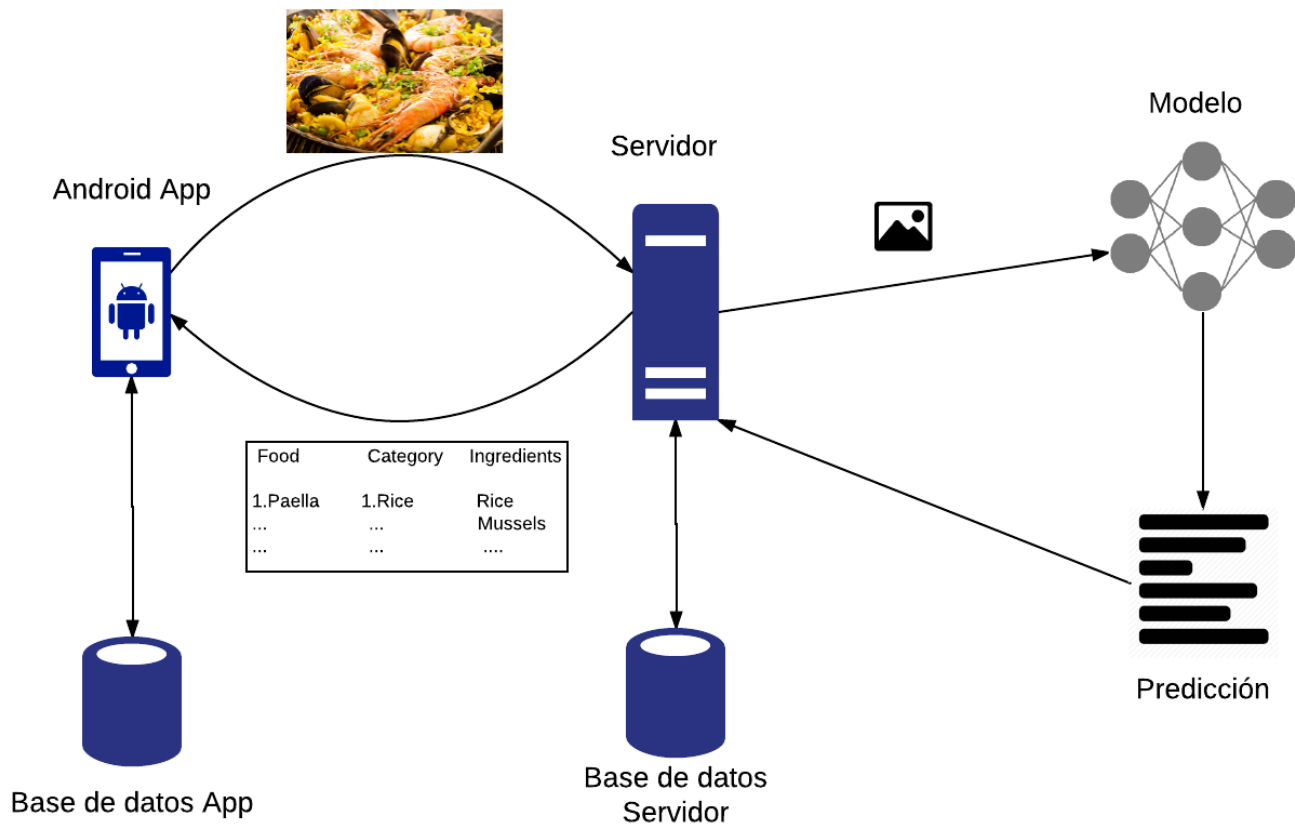


Figura 5: Arquitectura física del sistema

La aplicación envía una imagen al Servidor, que procesará gracias a los modelos de reconocimiento implementados. Esto generará una predicción, que será enviada a la aplicación. Este contexto no es la única situación bajo la que hay comunicación entre la aplicación y el servidor, pero es la más importante. En la sección 3.3 se analizarán con detalle todas estas comunicaciones. Tanto la aplicación como el servidor cuentan con una base de datos. En el primer caso, los datos serán

almacenados están más enfocados al rendimiento y experiencia del usuario dentro del contexto de la aplicación (valores nutricionales diarios, diario de comidas, etc). En cambio, en el caso del servidor, los datos son más genéricos, prestando gran atención al almacenamiento documentado de todas las imágenes que se procesan.

### 2.2.2. Diagrama de Clases de la aplicación

En la [Figura 6](#) se presenta el diagrama de clases de la aplicación. En este diagrama está el flujo principal de la aplicación representado por las conexiones entre las diferentes *activities*. En algunas de ellas el usuario puede realizar alguna acción adicional (por ejemplo añadir objetivo en *GoalActivity*) llevada a cabo en un *dialog*. Por otro lado, se pueden observar las comunicaciones con clases. En *ConfirmFoodActivity* el usuario consulta toda la información de la receta propuesta, y modifica si lo cree necesario la receta. Además, la información nutricional se actualiza según el usuario realice las modificaciones. Esta información se obtiene gracias a las clases *IngredientsInfo* y *RecipesInfo*. Una vez el usuario ha confirmado todas las opciones, se envía la información a la base de datos a través de la clase *ImageDataBase*. Se ha representado la librería *Volley*, que permite la comunicacuón rápida y eficaz con el servidor, utilizada en las clases *CaptureActivity* (para realizar el reconocimiento de la imagen) y *ConfirmFoodActivity* (para enviar la información confirmada por el usuario). Desde las secciones de estadísticas y objetivos también existe una comunicación con la base de datos (clase *ImageDataBase*) para obtener o modificar información acerca de las imágenes y los objetivos.



### 2.2.3. Bases de Datos de la aplicación y el servidor

Se analizarán a continuación los diseños de las bases de datos del sistema. Se han realizado diagramas de Entidad Relación, que permiten ver las entidades que se guardarán (se generarán tablas para cada entidad) y las relaciones entre ellas. Especificando las multiplicidades de las relaciones se podrá entonces proceder al paso a tablas de un modo preciso manteniendo estas relaciones.

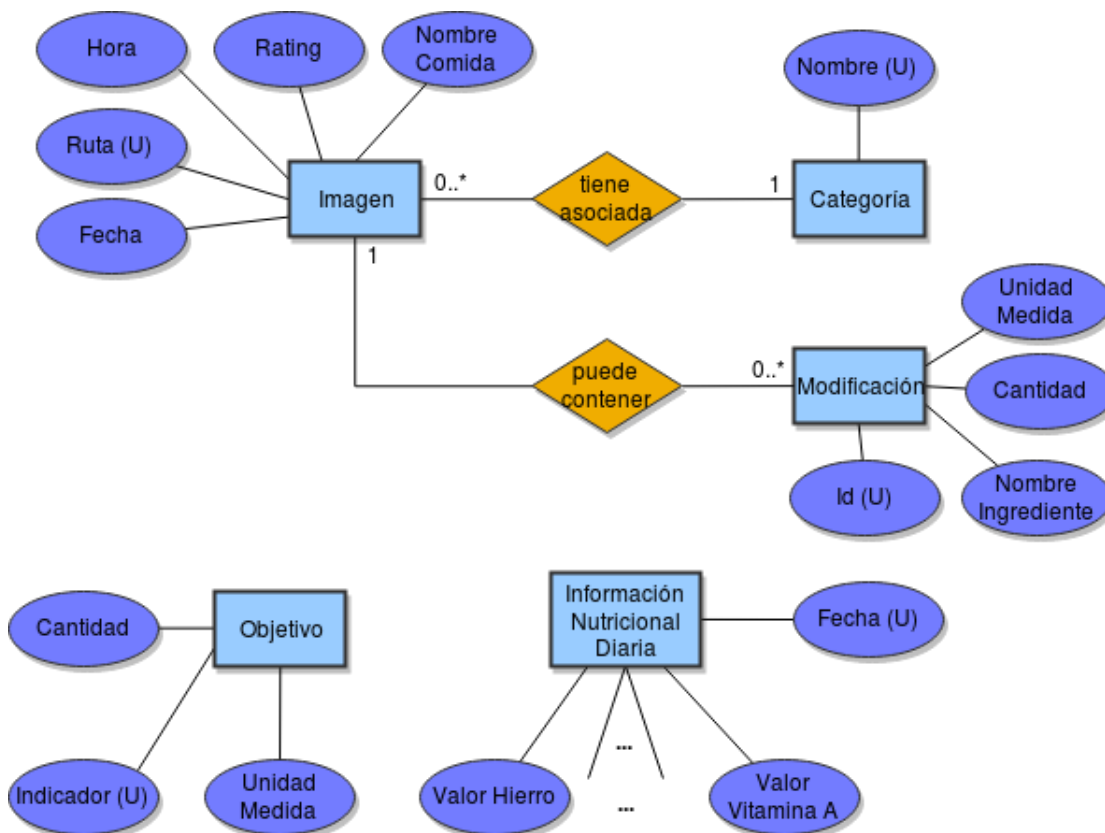


Figura 7: Diagrama Entidad Relación de la Base de Datos de la Aplicación

En la [Figura 7](#) se representa el diagrama de Entidad Relación de la base de datos de la aplicación. En primer lugar se tiene la entidad Imagen, que es el principal elemento de esta estructura de Base de datos. Una vez el usuario hace la imagen, esta se guarda automáticamente en una localización (ruta) generada. No obstante, la información de la Imagen no se guarda en la Base de Datos hasta que el usuario no ha confirmado el plato (entre los 5 propuestos). Está implementado el campo *rating* en la tabla de imágenes. Actualmente no se le está dando uso, pero un futuro se tiene la intención de dotar de un *rating* a la imagen, dependiendo lo sana que sea la comida.

Por otro lado se tienen dos entidades ligadas con la entidad Imagen. Para cada imagen capturada por el usuario se ofrece una categoría de comida propuesta (por ejemplo, postre, verdura, etc). Una

vez el usuario ha confirmado la categoría correcta, se guarda esta información en la base de datos, asociada con la imagen correspondiente. En el paso siguiente a confirmar la comida y la categoría, el usuario puede ver la receta propuesta y modificar los ingredientes. Estas modificaciones se guardan de modo genérico en la base de datos. Se guarda el nombre del ingrediente, junto con la unidad de medida usada y la cantidad. En un principio la unidad siempre será gramos, pero en un futuro se ofrecerá la selección de contar con otras unidades (por ejemplo 'cucharada' para ingredientes como aceite, vinagre, etc). Si la modificación ha consistido en borrar un ingrediente, se considera una modificación con un valor de 0 como cantidad. Si en cambio, se ha editado o añadido un ingrediente, se considera el valor correspondiente.

EatLog permite al usuario acceder a las comidas que ha registrado. Para mostrar la receta de cada comida, se procesa la receta genérica propuesta por defecto combinado a las modificaciones guardadas correspondientes a dicha comida (si las hay), siendo las modificaciones prioritarias a la receta genérica. Es decir, en el supuesto que una imagen corresponda a un plato con una receta que contiene un ingrediente que ha sido borrado por el usuario, esta información está almacenada en la entidad Modificación, y es priorizada ante la propuesta genérica de receta.

Se tiene por otro lado la entidad Objetivo, con los atributos cantidad, unidad de medida e indicador. Se ha considerado este último atributo como único, provocando que no pueda existir más de un objetivo para el mismo indicador nutricional. El usuario puede crear, editar y eliminar objetivos.

Finalmente se ha considerado la entidad Información Nutricional Diaria. Contiene el atributo fecha, considerado como unitario ya que se contiene un único objeto de esta entidad por valor diferente de fecha, es decir, por día. Por otro lado, contiene los 26 atributos correspondientes a todos los valores de los indicadores nutricionales que registra EatLog. Esta entidad no estaba concebida en un principio. Al hacer los resúmenes diarios de cada indicador nutricional, se debían analizar las imágenes de ese día, la comida correspondiente a cada imagen, y de esta comida la receta generada. También se debía comprobar y aplicar las modificaciones de esta imagen por parte del usuario. Todo este proceso suponía un tiempo muy grande (aproximadamente 4 segundos en el contexto de la aplicación EatLog). Por tanto se decidió generar automáticamente esta información. Cada vez que el usuario confirma el registro de una comida (después de gestionar los ingredientes), se aumentan los valores nutricionales de dicha comida a los índices diarios. Con este cambio, se notó una gran mejoría del rendimiento (el tiempo de respuesta se redujo a menos de 1 segundo, correspondiente únicamente a la creación de los gráficos).

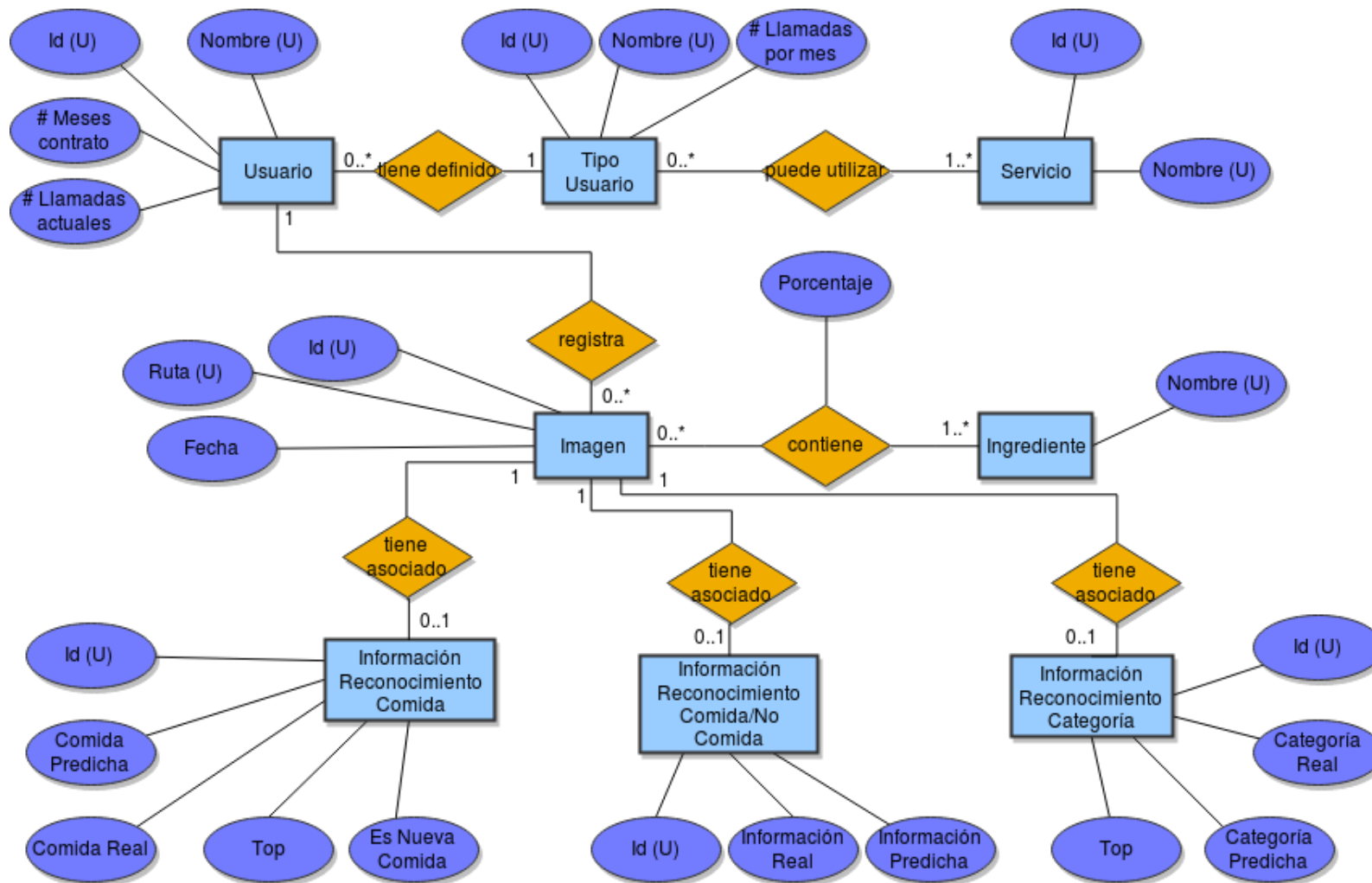


Figura 8: Diagrama Entidad Relación de la Base de datos del servidor

En la [Figura 8](#) se representa el diagrama de Entidad Relación de la base de datos del servidor. En primer lugar se ha concebido la entidad usuario. Este diagrama se ha diseñado teniendo en cuenta como concepto de usuario a los propios usuarios de la aplicación y a las entidades interesadas en utilizar los algoritmos de reconocimiento de imagen de EatLog sin la interfaz de la aplicación. El usuario tiene un nombre, la cantidad actual de llamadas y los meses de contrato(servirá para los usuarios externos de la aplicación). Por otro lado, el usuario se relaciona con el tipo de usuario. Se pueden considerar diferentes tipos de usuarios, todos con diferentes requisitos dependiendo de los reconocimientos que deseen aplicar a sus imágenes. Cada uno de estos reconocimientos es llamado Servicio en esta organización. Se tiene una relación entre el tipo de usuario y los servicios de reconocimiento que puede utilizar. Actualmente, se cuenta con el tipo de usuario 'admin' y 'test', ambos con un amplio número de llamadas por mes y todos los servicios de reconocimiento disponibles.

Por otro lado, se tiene la entidad Imagen. Toda imagen ha sido registrada por un usuario. Se han considerado 3 entidades diferentes para la información de los reconocimientos actuales (Comida/No comida, Categoría y Comida), todas ellas asociadas a una imagen. Ha sido conveniente separar estas tablas ya que cada reconocimiento tiene una información relevante diferente. En el contexto de reconocimiento de comida, es interesante saber en qué puesto (dependiendo de la probabilidad) se ha predicho la comida que corresponde a la imagen, lo que se denomina como *top*. Se guarda siempre la comida que predijo el reconocimiento con mayor probabilidad, la comida que corresponde realmente y si esta comida es nueva en la base de datos de EatLog. Esta información podrá ser aprovechada cuando el usuario de los servicios de reconocimiento dota de *feedback* al proceso. En el caso particular de la aplicación, el diseño está concebido de un modo que lo cumple. Una vez el usuario de la aplicación confirma la comida de la imagen dentro de las comidas propuestas, se guarda toda la información (predicha y real) correspondiente a los diferentes reconocimientos.

Finalmente se considera la entidad Ingrediente. Esta tabla está enfocada al contexto de la aplicación y el concepto de tener guardados los ingredientes que un cierto usuario suele dar como válidos para cada receta. Se relaciona cada imagen con los ingredientes que se han confirmado que corresponden a la imagen. Esta relación contiene el valor del porcentaje, que corresponde al porcentaje que da el reconocedor de ingredientes para cada ingrediente en concreto.



### 3. Herramientas e Implementación

#### 3.1. Reconocimiento de comida a través de *Deep Learning*

##### 3.1.1. Tecnología de reconocimiento, redes neuronales convolucionales

EatLog utiliza el Aprendizaje Profundo (Deep Learning) para llevar a cabo el reconocimiento de imágenes. Esta tecnología se compone de un conjunto de algoritmos que pertenecen a la clase aprendizaje automático (Machine Learning).

En concreto, el sistema utiliza la tecnología de las Redes Neuronales Convolucionales (Convolutional Neural Networks).

La neurona es una célula especializada y caracterizada por poseer una cantidad indefinida de canales de entradas llamados dentritas y un canal de salida llamado axón (ver [Figura 9](#)). Las dentritas operan como sensores que recogen información de la región donde se hallan y la derivan hacia el cuerpo de la neurona que reacciona mediante una sinapsis que envía una respuesta hacia el cerebro, esto en el caso de los seres vivos.

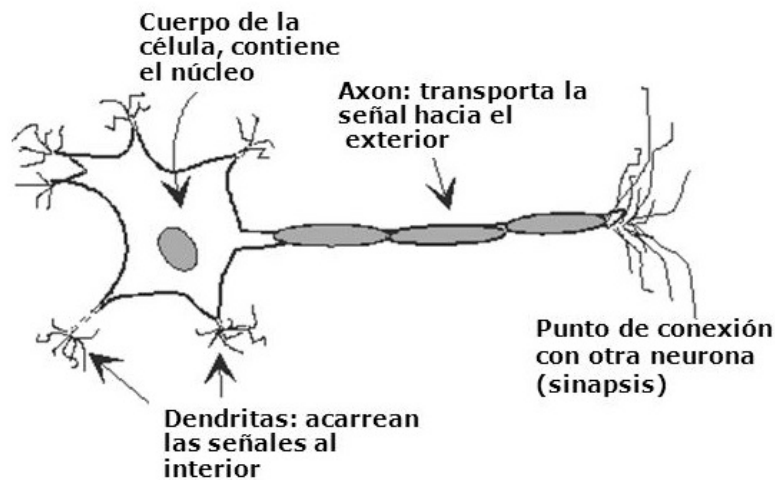


Figura 9: Modelo biológico de una red neuronal

El modelo matemático más simple de una neurona es un perceptrón (ver Figura 10). Una neurona sola y aislada carece de razón de ser. Su labor especializada se torna valiosa en la medida en que se asocia a otras neuronas, formando una red. Normalmente, el axón de una neurona entrega su información como 'señal de entrada' a una dendrita de otra neurona y así sucesivamente. El perceptrón que capta la señal en adelante se extiende formando una red de neuronas. Las características del perceptrón están determinadas por un vector de pesos reales y un umbral. El valor de salida, dependiendo si el valor del producto escalar del vector de pesos reales y el vector de entrada es superior al umbral.

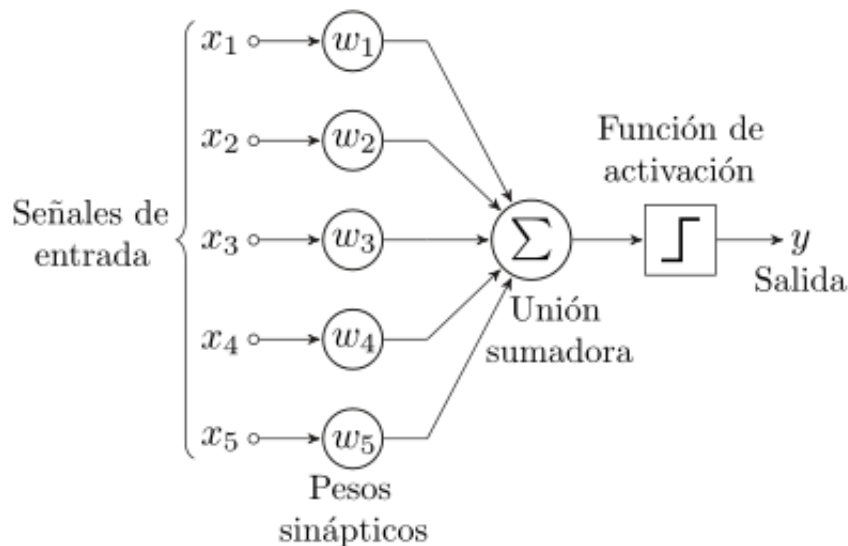


Figura 10: Modelo de un perceptrón

Las redes neuronales son un modelo computacional basado en un gran conjunto de unidades neuronales simples. Cada unidad neuronal está conectada con muchas otras y los enlaces entre ellas pueden incrementar o inhibir el estado de activación de las neuronas adyacentes. Estos sistemas aprenden y se forman a sí mismos. La propagación hacia atrás (backpropagation) es un algoritmo de aprendizaje supervisado donde se utiliza la estimulación hacia adelante para restablecer los pesos de las unidades neuronales, en combinación con una formación en la que se conoce el resultado correcto.

Una red neuronal convolucional es un tipo de red neuronal. La aplicación de esta red es realizada en matrices bidimensionales, por lo que es muy efectiva para tareas de visión artificial, como en la clasificación y segmentación de imágenes. En la fase de extracción de características, las neuronas sencillas de un perceptrón son reemplazadas por procesadores en matriz que realizan una operación sobre los datos de imagen 2D que pasan por ellas, en lugar de un único valor numérico. El operador que se aplica, llamado convolución, tiene el efecto de filtrar la imagen de entrada con un núcleo previamente entrenado que se va adaptando durante el entrenamiento. Esto transforma los datos de tal manera que ciertas características (determinadas por la forma del núcleo) se vuelven más dominantes en la imagen de salida. Después de una o más fases de extracción de características, los datos llegan a la fase de clasificación. Para entonces, los datos han sido depurados hasta una serie

de características únicas para la imagen de entrada, y es ahora la labor de esta última fase el poder clasificar estas características hacia una etiqueta u otra, según los objetivos de entrenamiento. En esta fase, las neuronas funcionan de manera análoga a las de un perceptrón.

### 3.1.2. Resultados

En el servidor se ejecutan varios reconocimientos de imagen. Las redes neuronales que se ocupan de ello están previamente inicializadas con todos sus parámetros, preparadas para procesar imágenes. Se pueden listar los diferentes tipos de reconocimiento:

#### COMIDA/NO COMIDA

El primer y más básico reconocimiento es la predicción sobre si la imagen corresponde a comida. En esta red la capa de salida de la red neuronal contendrá 2 neuronas. Los resultados son de un 95,64 % de acierto. [14]

#### TIPO DE COMIDA

Los algoritmos ejecutan el procesamiento de una imagen, teniendo como resultados unas posibles categorías con su probabilidad asociada. Esta red reconoce entre 200 categorías diferentes, 101 de ellas procedentes de Food-101 [9] y las 99 restantes proceden de la descarga de imágenes de EatLog. En el contexto de EatLog, se extrae la información de las 5 categorías más probables. Aquí, los resultados son de un 72,76 % de acierto considerando la métrica *Top1* y 90,44 % de acierto considerando la métrica *Top5*

#### CATEGORÍA

Se consideran grandes categorías de comidas, 11 en concreto. Por tanto, la capa de salida de la red neuronal contendrá 11 neuronas. El reconocimiento devuelve las categorías ordenadas por mayor probabilidad. Las 11 categorías que se reconocen son *Pan*, *Productos diarios*, *Postres*, *Huevo*, *Comida frita*, *Carne*, *Noodles/Pasta*, *Arroz*, *Pescado*, *Sopa* y *Verduras/Frutas* [15]. Los resultados del reconocimiento son de un 94,35 % de acierto.

#### INGREDIENTES

Se reconocen los ingredientes que componen una imagen. El algoritmo devuelve una lista de ingredientes con una probabilidad asociada. En este reconocimiento es más complejo llegar a una exactitud de los ingredientes reales de la imagen. Por ello, por ahora no se usa explícitamente este reconocimiento en el contexto de la aplicación EatLog. Esta información, no obstante, podrá ayudar en trabajos futuros a elegir la receta (es decir, la combinación de ingredientes) que mejor se adapte a la imagen analizada. Los resultados son de 53,43 % si nos referimos a *Precision*, 42,77 % en *Recall* y 47,51 % en *F-Measure*.

## 3.2. Aplicación

La aplicación EatLog ha sido desarrollada para dispositivos móviles con Android. Android es un sistema operativo basado en el núcleo Linux. Fue diseñado principalmente para dispositivos móviles con pantalla táctil, como teléfonos inteligentes, tablets y también para relojes inteligentes, televisores y automóviles. La arquitectura de Android se puede dividir en cinco capas (de mayor a menor nivel de abstracción) [4]:

**Aplicaciones:** En esta capa se encuentran por defecto las aplicaciones base, como puede ser navegador, contactos, entre otras. Todas las aplicaciones están escritas en lenguaje de programación Java.

**Marco de trabajo de aplicaciones:** Los desarrolladores tienen acceso completo a los mismos APIs del framework usados por las aplicaciones base.

**Bibliotecas:** Android incluye un conjunto de bibliotecas de C/C++ usadas por varios componentes del sistema. Una de ellas es SQLite, librería que usaremos en EatLog para implementar la base de datos local.

**Runtime de Android:** Android incluye un set de bibliotecas base que proporcionan la mayor parte de las funciones disponibles en las bibliotecas base del lenguaje Java. Cada aplicación Android corre su propio proceso, con su propia instancia de la máquina virtual Dalvik.

**Núcleo Linux:** Android depende de Linux para los servicios base del sistema como seguridad, gestión de memoria, gestión de procesos, pila de red y modelo de controladores.

Tal y como vemos en la [Figura 11](#) existen varias versiones de Android. La aplicación EatLog está disponible para dispositivos con versiones desde la 4.1.x a la 7.1, es decir, para las versiones con nombre Jelly Bean, KitKat, Lollipop, Marshmallow, Nougat. Los datos más recientes fechan del 5 Junio de 2017 [\[5\]](#). Según estos datos, EatLog cubre el 98,4 % de los dispositivos Android.

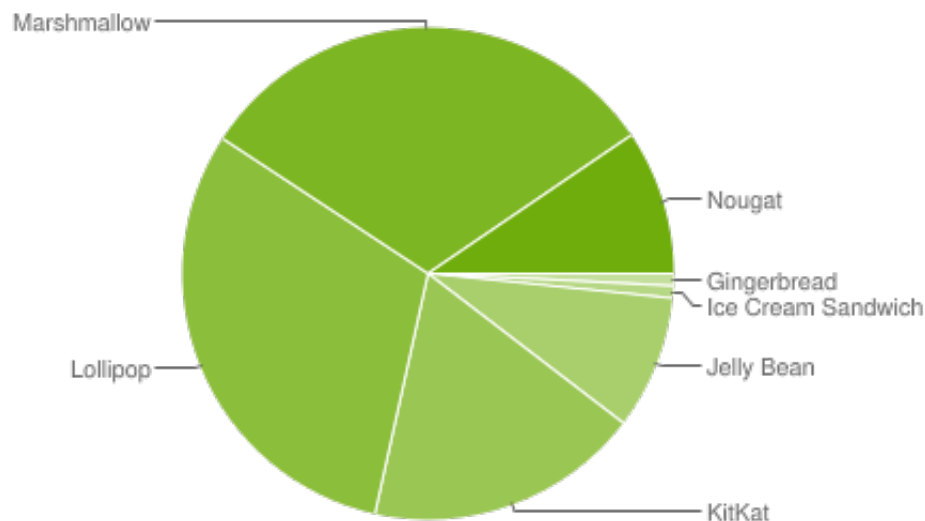


Figura 11: Reparto Mercado Android por Versiones [\[5\]](#)

En cuanto al reparto de los sistemas operativos que utilizan los dispositivos móviles, se tiene los siguientes datos que fechan de Febrero 2017[6], según la consultora Kantar World Panel [7]

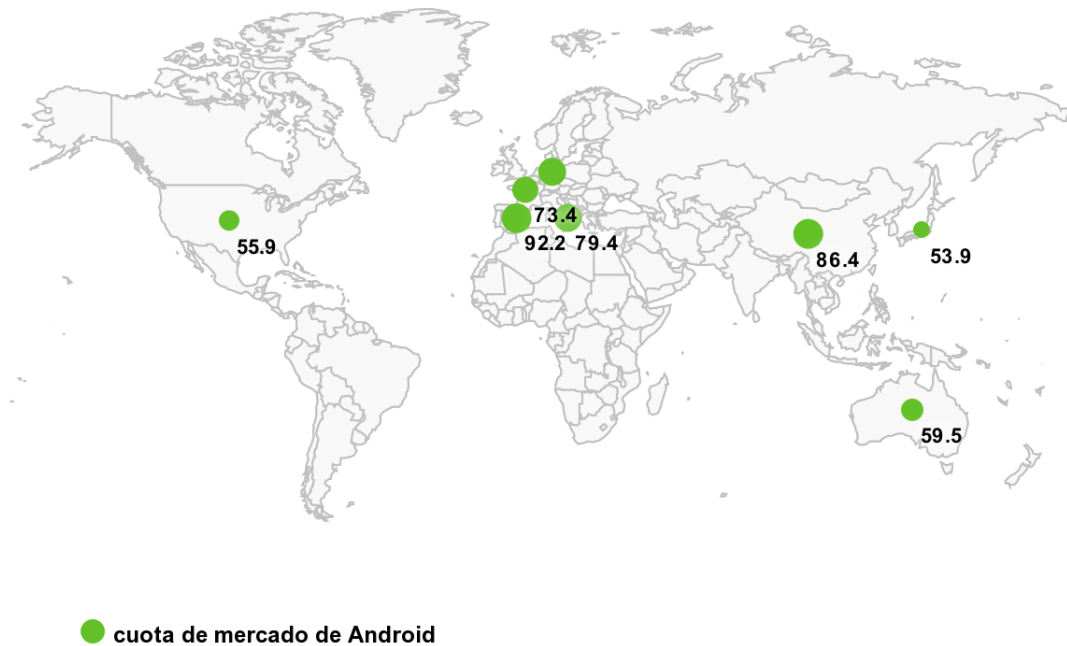


Figura 12: Cuota de mercado de Android

El sistema operativo Android representa en España el 92,2 % del mercado, un aumento de 2,2 puntos con respecto a febrero de 2016.

### 3.3. Servidor

Se ha implementado un servidor utilizando Flask, Python y SQLAlchemy para la base de datos del servidor. Se ha contado con un ordenador de la Universidad, dónde se ha reservado un puerto para el funcionamiento del servidor.

Tanto los algoritmos de reconocimiento como la base de datos (y su gestión) están implementados en este servidor, totalmente independiente de la aplicación para móvil. Con esto se ha querido dotar de adaptabilidad al proyecto y estar abiertos a futuras aplicaciones. Con esta lógica se han implementado unos métodos genéricos de acceso a los reconocimientos, y una estructura de base de datos que se adapta a un posible contexto futuro.

En particular, uno de los grandes objetivos de EatLog es incluir el reconocimiento automático en un asistente digital en forma de aplicación. Para ello es entonces necesaria la comunicación entre la aplicación y el servidor. La aplicación genera peticiones POST dentro del protocolo de

comunicación HTTP, gracias a la librería para Android llamada Volley [8]. Esta librería facilita y acelera las comunicaciones entre la aplicación y el servidor. Se representarán en forma de diagramas las diferentes comunicaciones entre la aplicación y el servidor.

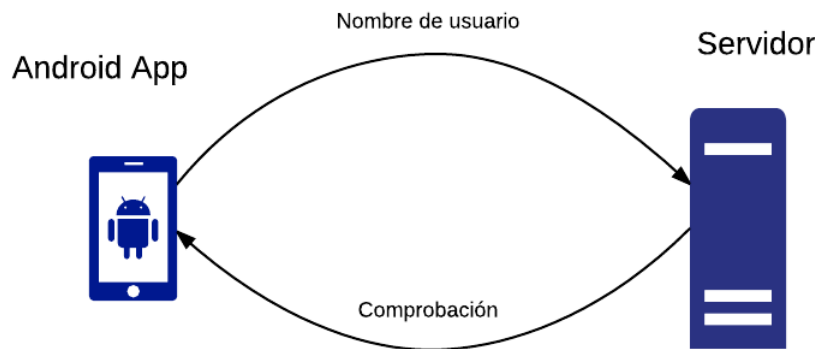


Figura 13: Envío de información sobre el usuario

La aplicación envía el nombre de usuario elegido en la primera ejecución de la aplicación. El servidor se ocupa de comprobar si dicho nombre de usuario existe y enviar dicha información a la aplicación. Si el nombre de usuario ya existe, se mostrará un mensaje informando sobre ello y se pedirá un nuevo nombre de usuario.

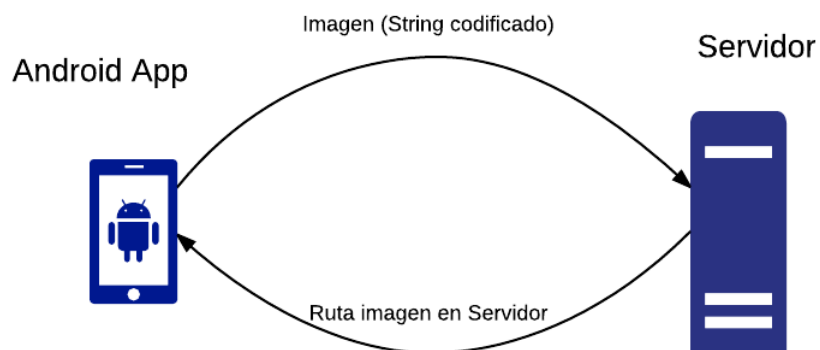


Figura 14: Envío de la imagen al servidor

Una vez se tiene la petición de reconocer una imagen por parte del usuario, se envía dicha imagen (codificada en String) al servidor. El servidor descodifica el string, genera la imagen asociada y la

guarda en una ruta específica que devuelve como información a la aplicación.

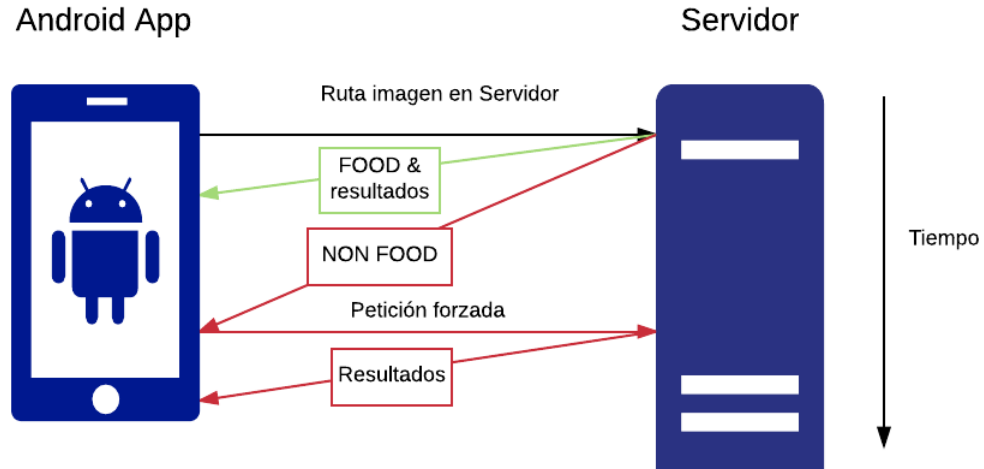


Figura 15: Envío de resultados desde el servidor

Tras la comunicación anterior, la aplicación envía al servidor la ruta de la imagen que se quiere reconocer. El servidor ejecuta entonces los algoritmos de reconocimiento, empezando por el correspondiente a Comida/No comida. Si el resultado es Comida, se ejecutan el resto de reconocedores y se envía toda la información en formato JSON a la aplicación. Si el resultado es 'No comida', no se ejecutan el resto de reconocedores y se envía un mensaje por parte del servidor conteniendo dicha información. En este punto, la aplicación pregunta al usuario si realmente esta imagen es comida (dando la opción de que realmente lo sea). Si el usuario confirma que no corresponde a comida, el flujo de comunicación acaba aquí. Sin embargo, si el usuario afirma que la imagen corresponde a comida, la aplicación envía de nuevo la ruta de la imagen en el Servidor, junto con un parámetro que fuerza la ejecución de todos los reconocedores. De este modo, el servidor cumple con la orden recibida y devuelve los resultados de los reconocimientos en formato JSON a la aplicación.

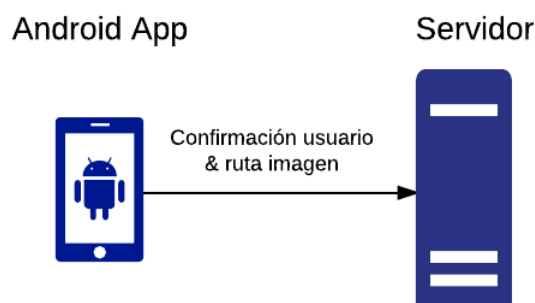


Figura 16: Envío de confirmación desde la App

El usuario ha recibido una propuesta de varios platos y categorías (ordenados por probabilidad) con respecto a la imagen enviada. En este punto, debe elegir entre las opciones dadas o añadir un nuevo plato. Dependiendo del plato elegido, en la siguiente pantalla se muestra al usuario el listado de ingredientes y el valor nutricional del plato. Existe la posibilidad de personalizar los ingredientes, borrando, añadiendo o editando. Una vez el usuario ha confirmado los ingredientes, se envía al servidor toda la información que ha sido aportada (el plato confirmado, la categoría confirmada, y los ingredientes confirmados) junto con la ruta de la imagen. De este modo, el servidor puede recolectar todos estos datos, lo que resulta un punto importante del proyecto, tal y como se ha explicado con anterioridad.

### 3.4. Construcción de los DataSets

#### 3.4.1. Descarga de Imágenes de Comida

Las imágenes utilizadas para entrenar los algoritmos de reconocimiento surgen de dos bases de datos. Una de ellas, es Food-101 [9], que contiene 101 categorías con 1000 imágenes por categoría, es decir, un total de 101000 imágenes. Por otro lado, se cuenta con un trabajo previo de creación de la base de datos de imágenes FoodCat [10], compuesta de 115 categorías con al menos 100 imágenes por categoría.

Uno de los objetivos es ampliar la cantidad de imágenes por categoría. Un aspecto importante es la descarga de imágenes en alta resolución.

Se han descargado imágenes de Google y de Instagram. Las necesidades de automatizar la interacción con el navegador impulsaron el uso del WebDriver de Selenium.

Selenium es un entorno de pruebas de software para aplicaciones basadas en la web, enfocado en la prueba automática del funcionamiento de las webs, siendo muy útil para el testing. Selenium WebDriver es una herramienta que permite una interacción libre con la web. [11]

En un principio se intentó llegar a cubrir toda la variedad mundial de comidas. Se implementó un script que automatizó la selección de comidas, organizadas por tipos de comida [12]. El total de comidas obtenido es de aproximadamente 7800. Se comenzó la descarga de imágenes de Google y Instagram de estas comidas, llegando a tener las imágenes correspondientes a aproximadamente



2000 categorías.

Dada la complejidad y dificultad de etiquetar y entrenar sobre tantas categorías, se decidió marcar objetivos prioritarios de comidas locales.

Se ha contactado y tenido relaciones a lo largo del proyecto con nutricionistas que han guiado la elección de categorías a descargar, priorizando las más comunes en el contexto local.

#### 3.4.1.1 Descarga de imágenes procedentes de Google

Las imágenes hasta ahora descargadas a través de Google tenían una resolución relativamente pequeña, alrededor de 250x250. Los enlaces de las imágenes que aparecen al obtener el código html de la web de 'Google Images' corresponden a imágenes de baja resolución. Se expondrá a continuación la explicación de la implementación de la descarga de las imágenes en alta resolución, apoyada en el diagrama de flujo de la [Figura 17](#).

Se investigó el modo de conseguir los enlaces de las imágenes en alta resolución, sabiendo que esto es accesible para cualquier usuario. Se implementó un script que recoge los enlaces de las imágenes en alta resolución de Google. Se concebirá como agente la persona ficticia que se encarga de llevar a cabo todas las acciones automatizadas por el script. En primer lugar, el agente debe desplazarse hacia abajo, con el fin de cargar más imágenes. Una vez se han cargado aproximadamente 500 imágenes, el agente pulsa un botón que permite cargar más imágenes. Se desplaza hacia abajo nuevamente, hasta llegar a un máximo de 1000 imágenes cargadas. En este punto se tiene acceso a todos los enlaces en baja resolución.

Para conseguir el enlace en alta resolución, se debe simular el clic de las imágenes mostradas (en tamaño reducido), de este modo se amplía la imagen y se muestra el botón de 'Guardar Imagen', donde está asociada la ruta de la imagen en alta resolución. Al clicar en cada imagen previa, también se tiene acceso al enlace en alta resolución de las dos imágenes contiguas. De este modo, se reduce el tiempo de obtención de enlaces a un tercio del original.

Al clicar en las imágenes previas de Google, se modifica la url del navegador, agregando un identificador al final. Estos identificadores se pueden obtener previamente, en la página principal de Google. El agente guarda estos identificadores una vez ha conseguido que todas las imágenes previas estén mostradas. Ahora, se generan peticiones a la web añadiendo el identificador de cada imagen (en realidad de una imagen de cada tres). A continuación se genera la petición de volver a la página principal. Se repite esto, hasta conseguir los enlaces en alta resolución de todas las imágenes.

Teniendo la lista de los enlaces, se puede proceder a la descarga de las imágenes. La descarga de estas imágenes se implementa en threads, lo que permite un rendimiento mayor. Estos threads comparten la estructura de cola que contiene los enlaces de las imágenes en alta resolución. Además, se guarda en un documento una asociación entre la ruta generada de la imagen guardada y el enlace de dicha imagen. Esta decisión ha sido tomada por argumentos de seguridad.

En la [Figura 18](#) se pueden ver capturas de imagen de *Google*. La imagen de la izquierda corresponde a la pantalla de imágenes de Google al realizar una búsqueda. La imagen de la derecha corresponde a la misma página una vez se clicca en alguna de las imágenes previas. Es en dicha pantalla donde se consiguen los enlaces en alta resolución de la imagen ampliada y las dos imágenes contiguas a ella.

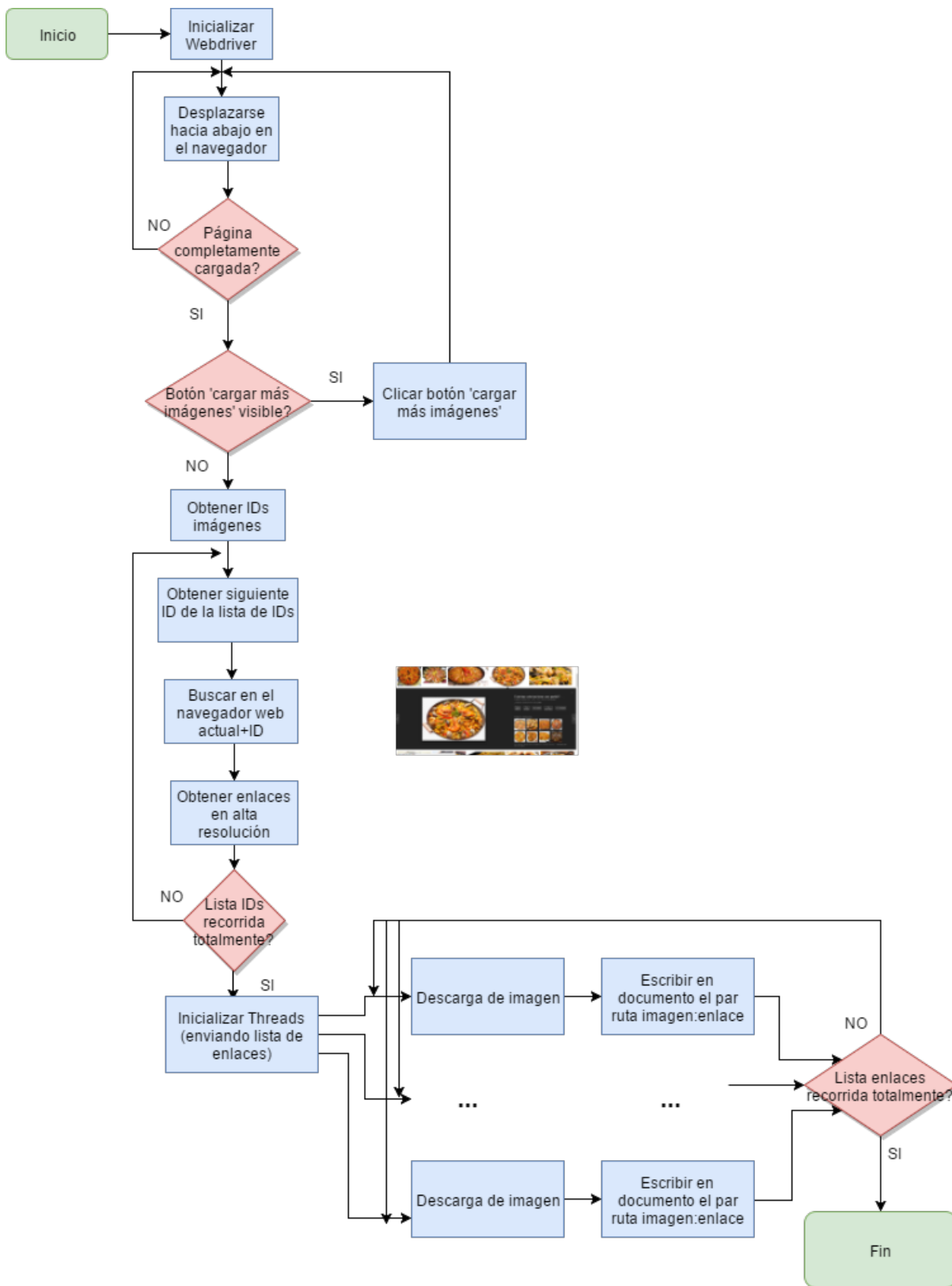


Figura 17: Diagrama de Flujo del código correspondiente a la descarga de imágenes de Google

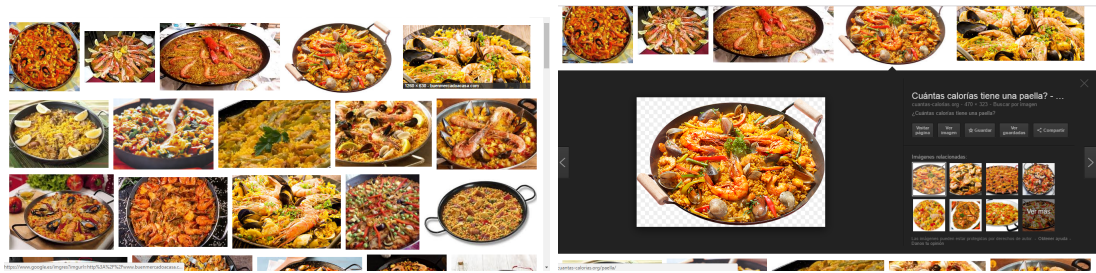


Figura 18: Capturas de imagen de Google

### 3.4.1.2 Descarga de imágenes procedentes de Instagram

Más adelante se hablará de cómo se han etiquetado y filtrado todas estas imágenes descargadas. Después de filtrar las imágenes descargadas con Google, se han obtenido unas 400 imágenes de media por categoría, siendo una cantidad muy baja para obtener buenos resultados de reconocimiento. Se decidió aplicar los conceptos de navegación automática ya expuestos y desarrollados para descargar imágenes de la red social Instagram. En la [Figura 19](#) se presenta un diagrama de flujo que indica el funcionamiento general del código.

Se inicializa la instancia del WebDriver, en particular se está usando el navegador Google Chrome. Se crea un usuario de Instagram, con el que autenticarse para buscar las imágenes. Se añade a la url del navegador como parámetro de búsqueda ('q' de query) el nombre de la comida. Se accede entonces a la página dónde se muestran todas las imágenes que contienen como 'tag' el nombre que estamos buscando. En esta ocasión el enlace de la imagen en alta resolución es visible directamente en esta propia página, lo que facilita la descarga. No obstante, únicamente se muestran las primeras imágenes por lo que se ejecuta el clic virtual del botón de cargar más imágenes. Esto solo es necesario una vez, ya que para las sucesivas cargas de imágenes es suficiente con desplazar el navegador hacia el final de la página.

En una primera versión, se decidió actuar de forma análoga que con Google, es decir, se obtienen en primer lugar todos los enlaces de las imágenes y a continuación se descargan. Además de este modo se pueden utilizar Threads para las descargas. Surgió un problema, ya que se descargaban muchas imágenes seguidas, lo que captó la atención de Instagram, que invalidaba la IP temporalmente. Por otro lado, en el navegador se tenía que esperar unos ciertos segundos entre cada desplazamiento hacia abajo de la página cuando se busca que se carguen las imágenes.

Surgió entonces la opción de descargar las nuevas imágenes visibles correspondientes a cada desplazamiento del navegador hacia abajo. Es decir, con cada desplazamiento se cargan 15 nuevas imágenes, que se descargan en alta resolución. De este modo, existe un tiempo entre descargas que permite no ser bloqueados, y se aprovecha el tiempo de espera entre cada desplazamiento del navegador para descargar imágenes.

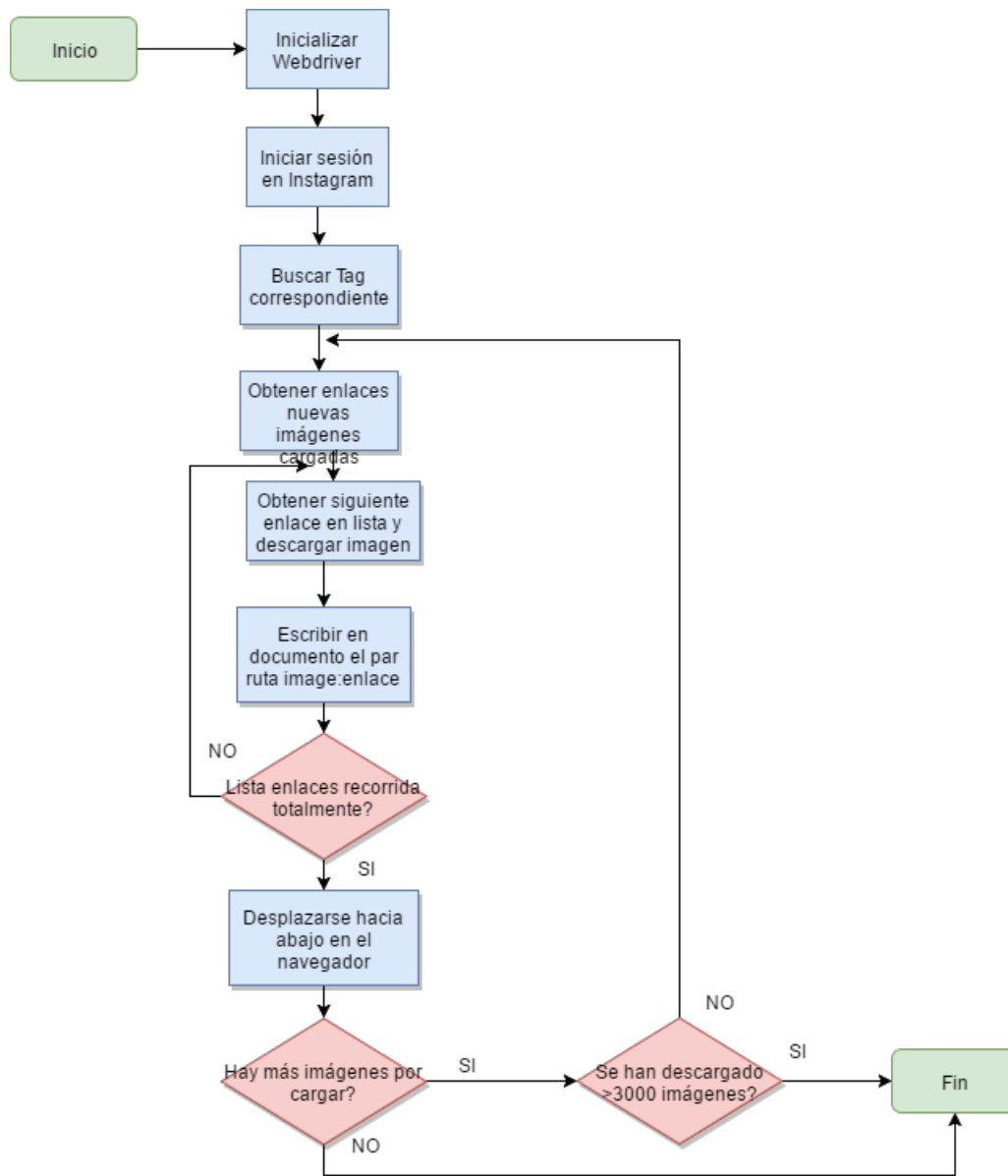


Figura 19: Diagrama de Flujo del código de descarga de imágenes de Instagram

#### 3.4.1.3 Curación de las imágenes usando la herramienta web de etiquetado

El trabajo de validar las imágenes descargadas tiene una gran importancia. La descarga ha sido automatizada por lo que no se ha verificado que las imágenes sean correctas, y correspondan con la búsqueda realizada. Un miembro del equipo implementó en el servidor una herramienta de etiquetado en formato web. Antes de proceder al etiquetado de las imágenes, se han filtrado las

imágenes descargadas con la aplicación de un reconocimiento de comida/no comida. Esta interfaz web (Figura 20) ha permitido llevar a cabo el etiquetado de las imágenes de manera organizada y eficaz por parte de varios miembros de EatLog.



Figura 20: Captura de la Web utilizada para el etiquetado de imágenes

#### 3.4.1.4 Resultados

EatLog cuenta actualmente con una base de datos de 99 categorías que contienen un mínimo de 600 imágenes en alta resolución por categoría. En la Tabla 10 se puede ver la cantidad de imágenes de media para cada una de estas 99 categorías. Como se puede apreciar, aproximadamente un tercio de las imágenes descargadas inicialmente son consideradas como válidas.

Imágenes por categoría tras la descarga	Imágenes por categoría tras aplicar los filtros del reconocimiento Comida/No Comida	Imágenes por categoría tras etiquetar gracias a la herramienta web.
2213	1793	796

Tabla 10: Datos sobre la cantidad de imágenes antes y después del filtrado y etiquetado

Algunas de las categorías propuestas han tenido que ser desechadas, dada la dificultad de conseguir imágenes. Actualmente, EatLog está en proceso de filtrar, etiquetar y entrenar con nuevas imágenes ya descargadas.

Siguiendo las indicaciones de profesionales del sector de la nutrición, se descargaron imágenes correspondientes a Frutas y Verduras (88 categorías).

Por otro lado, se han descargado imágenes de las 115 categorías correspondientes a FoodCat [10]. Se está procesando actualmente una comparación entre las imágenes descargadas de Google en alta resolución con las imágenes originales de FoodCat en menor resolución. De este modo, se puede reaprovechar el trabajo de filtrado de imágenes llevado a cabo en la creación de la base de datos FoodCat. Teniendo como objetivo 600 imágenes por categoría, se calcula que se puedan materializar entre 40 y 50 categorías de las 115 originales.

### **3.4.2. Construcción de los Datasets de Recetas e Ingredientes**

#### **3.4.2.1 Información a descargar**

El objetivo es obtener la información nutricional de las comidas con la mayor exactitud y adaptabilidad posible. Se busca obtener el listado de ingredientes que componen las diferentes recetas de una comida, con la medida asociada. La información nutricional de cada receta es interesante, pero es preferible generarla que obtenerla. Por lo tanto, se necesita la información nutricional de cada ingrediente. De este modo, se podrán generar los valores de todos los indicadores nutricionales, y así adaptarse a las modificaciones de los ingredientes. En la [Figura 21](#) está representado un diagrama de flujo del código correspondiente a la descarga de esta información.

#### **3.4.2.2 Implementación**

Se descargan los datos gracias a la API de Edamam [13], una de las webs con mayor base de datos de recetas e ingredientes. Se ofrecen dos servicios, el servicio de Recetas y el servicio de Información Nutricional. En el primero, se envía el nombre de un plato y se recibe un listado con todas las recetas correspondientes. En el segundo servicio de la API, se envía un listado de ingredientes, y se obtiene la información nutricional asociada.

En primer lugar se obtienen las recetas (se puede acceder a las 300 primeras recetas propuestas para la comida que buscamos). Dentro de cada receta, además de los ingredientes, se tiene la información de cómo se lleva a cabo la receta, para cuántas personas es y su información nutricional. De todos estos datos, es interesante quedarse con la lista de ingredientes con las medidas asociadas y para cuántas personas es la receta.

Con estos datos, se genera un listado de todos los ingredientes contenidos en la primera receta de cada comida. En la primera versión de la implementación este paso fue sencillo. La API de Edamam devolvía información completa del ingrediente, es decir, el texto contenido en la receta (por ejemplo, 'queso manchego fundido') y el texto correspondiente al ingrediente real (en este caso 'queso'). De este modo, se tuvo en cuenta la información del ingrediente real para generar la base de datos de ingredientes. No obstante, en las últimas actualizaciones que se llevaron a cabo, ha habido un cambio en los resultados que devuelve la API de Edamam. Actualmente, se devuelve únicamente el texto correspondiente a la descripción del ingrediente en la receta. No es conveniente mantener esta información como una descripción genérica del ingrediente.

Se llama entonces al segundo servicio de la API de Edamam, que permite obtener el valor nutricional de un listado de ingredientes que componen una receta. En particular, se quiere obtener la información de cada ingrediente por separado, por lo que se deben hacer llamadas independientes a este servicio. El problema expuesto en el párrafo anterior se soluciona gracias a esta información. En una primera versión, se obtenían los nombres directos de los ingredientes, por lo que se ejecutaban llamadas a este servicio de la API y se guardaba la información nutricional. En la actualidad el objetivo y el procedimiento son parecidos, se ejecutan llamadas a la API enviando el texto del

ingrediente dentro de su receta. Edamam, para poder ofrecer la información nutricional de lo que se está pidiendo, tiene un sistema para asociar un texto de un ingrediente con un ingrediente de su base de datos. Este sistema no es totalmente eficaz, pero permite pasar por ejemplo de 'pimienta de cayena machacada y espolvoreada' a 'pimienta de cayena'. Ya se ha conseguido una mejor generalización del ingrediente. No obstante, las recetas continúan guardando como descripción de sus ingredientes el texto inicial. Por tanto, se recorre cada receta y se reemplazan el texto original por la escritura más genérica obtenida.

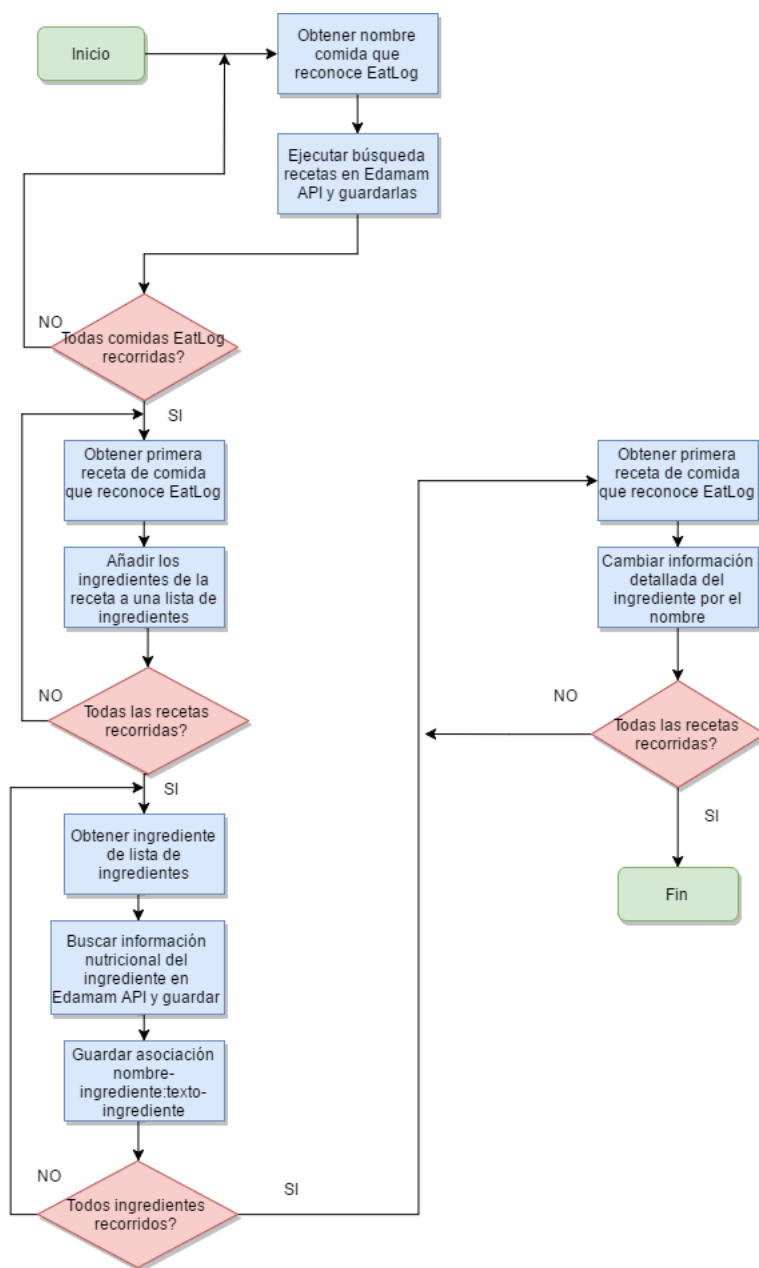


Figura 21: Diagrama de flujo del código de descarga de recetas e información nutricional de ingredientes



### 3.4.2.3 Resultados

Para cada ingrediente se tienen todos los valores nutricionales asociados por 100 gramos de dicho ingrediente (en total 26 indicadores nutricionales).

```
ingredientName": "garlic"  
  "nutrients":  
    "FE":  
      "quantity": 1.7,  
      "unit": "mg",  
      "label": "Iron"  
    "PROCNT":  
      "quantity": 6.36,  
      "unit": "g",  
      "label": "Protein"  
    etc
```

Figura 22: Ejemplo de la información nutricional almacenada para cada ingrediente.

Por otro lado, se tiene el listado de ingredientes de cada receta.

```
"foodName": ".apple+pie",  
  "ingredients": [  
    "name": "vanilla ice cream",  
    "weight": 50.0  
  
    "name": "whole milk",  
    "weight": 30.5  
  
    "name": ".apple pie, frozen",  
    "weight": 4.84375  
  ]
```

Figura 23: Ejemplo de la información almacenada de cada receta.

EatLog contiene recetas de 189 de los 200 platos que reconoce. En cada plato la cantidad de recetas varía, desde 10 hasta 300 (el máximo de recetas por plato que da Edamam). Esta información es importante, y se usarán las primeras recetas de cada plato. No obstante, ahora se está utilizando únicamente la primera receta. Agrupando los ingredientes diferentes de todas las primeras recetas, se ha descargado información nutricional de 897 ingredientes. Está en proceso la descarga de más ingredientes, sabiendo que este listado actual depende únicamente de la primera receta de cada plato. Actualmente, se tiene información nutricional de unos 2000 ingredientes. La versión de la aplicación EatLog cuenta entonces actualmente con 189 recetas y 897 ingredientes.

## 4. Manual de usuario

Aquí se mostrarán las pantallas de la aplicación, explicando el funcionamiento de cada una de ellas. En definitiva, se definirá el manual de usuario de EatLog, tal y como lo indica el título de la sección.

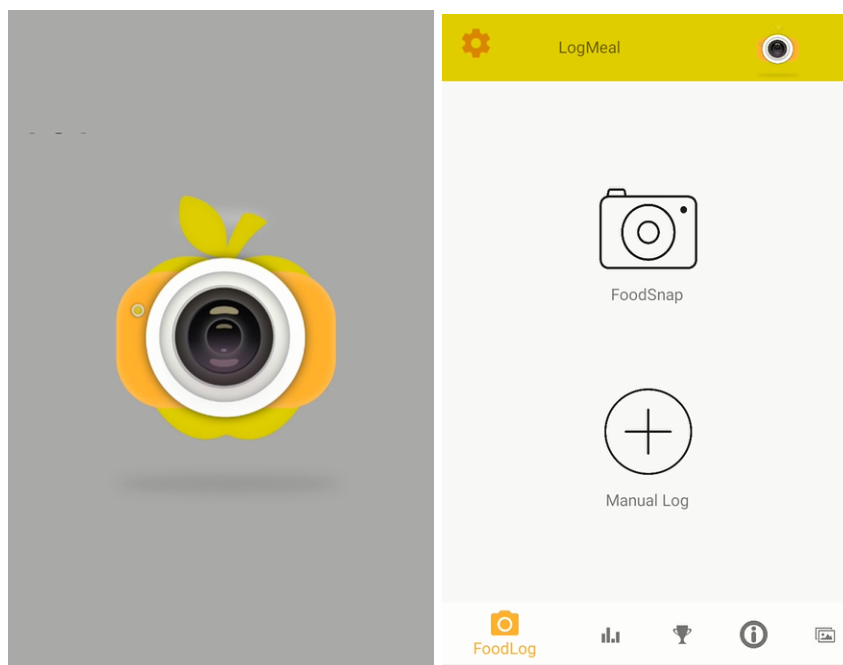


Figura 24: Pantallas de la aplicación: pantalla presentación y pantalla principal

La pantalla de presentación se ejecuta mientras se configuran los registros al iniciarse la aplicación. Actualmente, esto no requiere de mucho tiempo. A continuación se muestra la pantalla inicial, por defecto en la pestaña de *FoodLog*. El usuario puede aquí elegir registrar una comida realizando una foto en directo o registrarla de forma manual, ya sea con una imagen ya tomada anteriormente como introduciendo la comida sin imagen. Si el usuario clicca sobre el icono de la cámara, se muestra la pantalla de la [Figura 25](#). Si el usuario en cambio clicca en introducir manualmente una comida, se muestra la pantalla de la [Figura 26](#).



Figura 25: Pantalla de la aplicación: pantalla de cámara

Como se puede ver, la pantalla de la [Figura 25](#) corresponde a una aplicación externa del móvil que permite hacer una foto. El usuario puede realizar la foto, con un funcionamiento lógico e intuitivo. Al confirmar la foto realizada, se mostrará la [Figura 27](#).

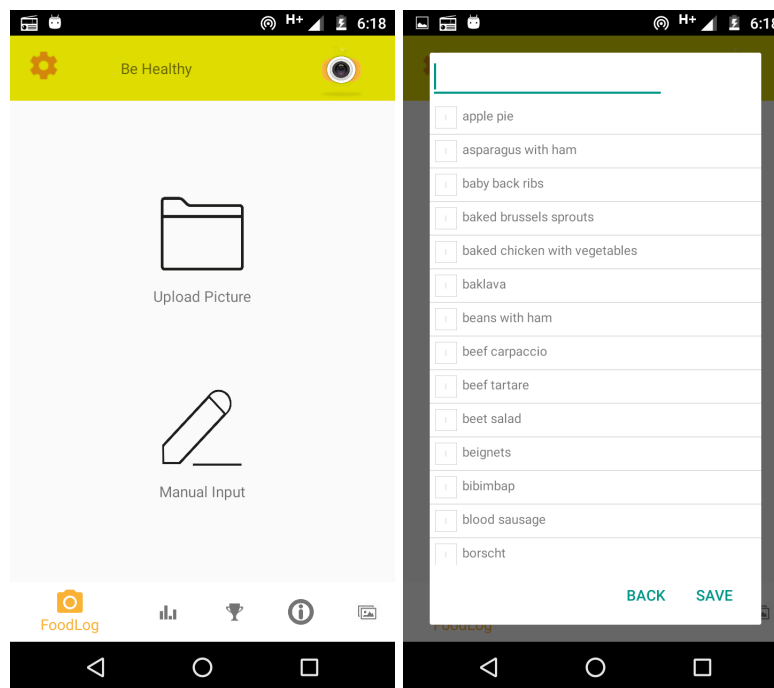


Figura 26: Pantallas de la aplicación: pantalla de entrada manual de comida y pantalla de escritura manual de comida

El usuario ha elegido la opción de introducción manual de la comida. Se muestra la pantalla (pantalla de la izquierda de la [Figura 26](#)) donde está las opciones de subir una imagen existente, o registrar una comida sin imagen. Si el usuario clicla la primera opción, la aplicación le envía a su directorio de imágenes donde puede elegir la imagen que desea añadir. Una vez la elija se mostrarán las pantallas de la [Figura 27](#). Si el usuario clicla la segunda opción, es decir, registrar una comida sin imagen, se le mostrará la pantalla de la derecha. En esta pantalla, se podrá escribir el nombre de una comida, o buscar entre los que reconoce EatLog. Al confirmar la elección, se mostrarán las pantallas de la [Figura 28](#).

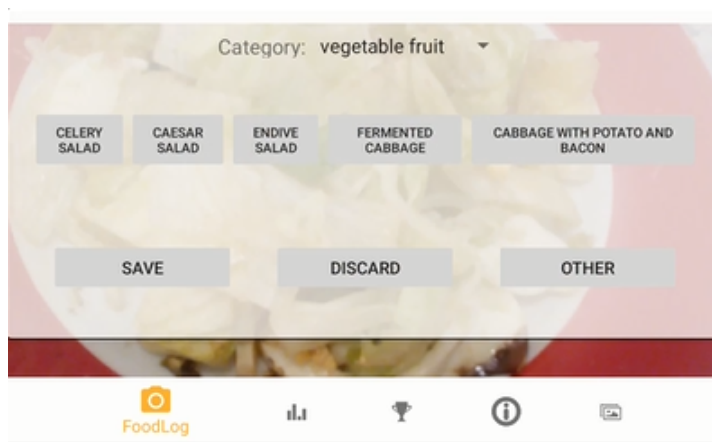


Figura 27: Pantalla de la aplicación: pantalla de resultados de reconocimiento de la imagen

En este manual no se muestra, pero si la imagen se reconoce como no correspondiente a comida, se muestra un mensaje preguntando si realmente la imagen es comida. Si el usuario lo confirma, el sistema lo procesa y se pasa a esta pantalla. Supone un filtro interesante por ambos motivos, el usuario no puede hacer un mal uso de la aplicación (intentando reconocer una mochila, por ejemplo) y por otro lado, previene si una imagen no ha sido bien tomada. Es decir, si el reconocimiento da como resultado no comida a una imagen que sí lo es, es muy probable que la imagen no haya sido bien capturada. En la pantalla de la [Figura 27](#) se mostrará la imagen de fondo (ya sea tomada en el momento o subida desde el teléfono) con la información de los reconocimientos. El usuario debe decidir que plato de los 5 propuestos es el que corresponde a la imagen, así como la categoría. Si ninguno de los 5 coincide, el usuario puede cliclar en *Other* y se mostrará la pantalla que hemos visto en la [Figura 26](#) (la imagen de la derecha). Una vez el usuario confirma las elecciones, se muestra la pantalla de la [Figura 29](#) si el usuario ha realizado la foto, o las pantallas de la [Figura 28](#) si el usuario ha subido la imagen.

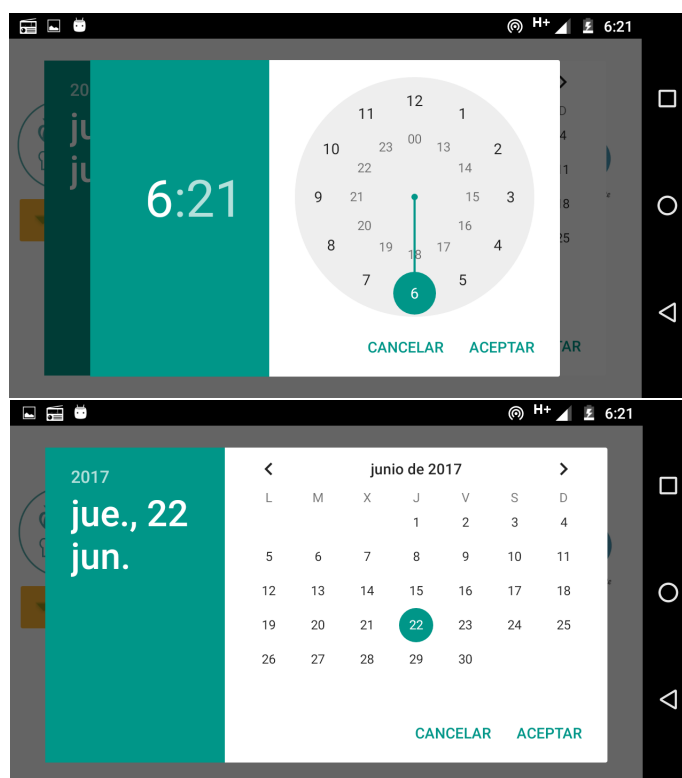


Figura 28: Pantallas de la aplicación: pantalla de elección de hora y pantalla de elección de fecha

Las pantallas de la [Figura 28](#) son mostradas cuando el usuario ha deseado registrar una comida de forma manual (subiendo imagen o escribiendo el nombre de la comida). En estas pantallas, se debe seleccionar la hora y fecha de la comida que se está registrando. Esta opción no se muestra cuando el usuario elige realizar la foto, ya que se da por hecho que la comida se está desarrollando en ese mismo momento. Una vez se confirma la fecha y hora, se muestra la pantalla de la [Figura 29](#).

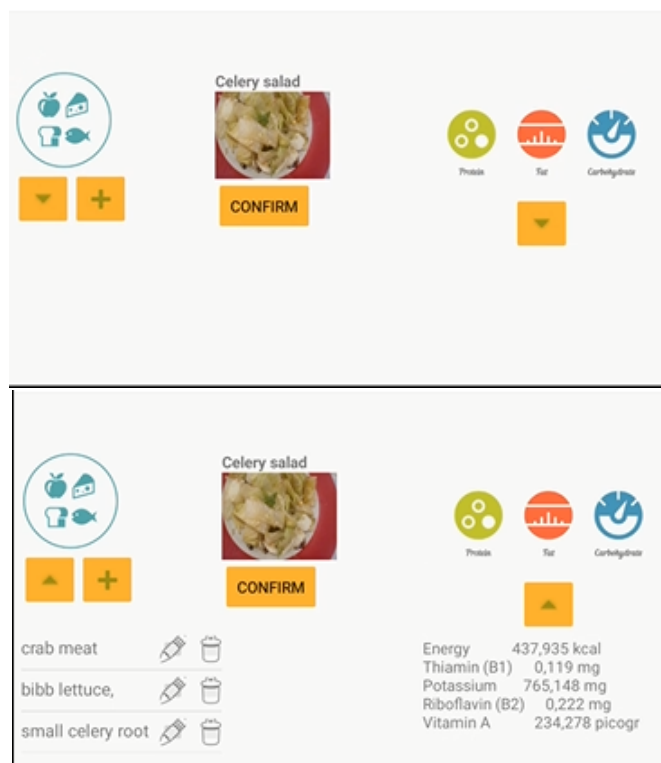


Figura 29: Pantallas de la aplicación: pantallas de la información nutricional y del la receta de la comida

En las pantallas de la [Figura 29](#) se ve como el usuario puede acceder a la información de las recetas y a la información nutricional (inicialmente escondidas). En el listado de ingredientes, se pueden editar y borrar cada uno de ellos, así como añadir un ingrediente nuevo. Si se clicca en el icono de añadir ingrediente, se mostrará la pantalla de la [Figura 30](#). Cualquier cambio en la lista de ingredientes se verá reflejado automáticamente en la lista de información nutricional. Si el usuario clicca en confirmar, se mostrará la pantalla de la [Figura 31](#).

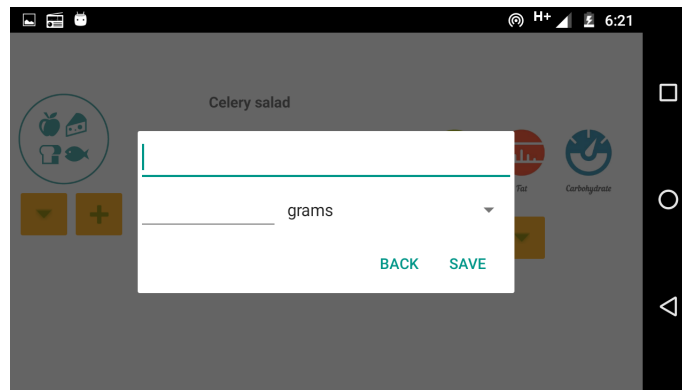


Figura 30: Pantalla de la aplicación: pantalla de añadir ingrediente

En la [Figura 30](#) se muestra la pantalla donde el usuario puede añadir un ingrediente. Hay un campo de texto donde se escribirá el nombre del ingrediente. Este campo está dotado de la opción de autocompletado, para favorecer la introducción de ingredientes ya existentes en la base de datos de EatLog. Una vez el usuario escribe el nombre del ingrediente, debe escribir la cantidad (en gramos por ahora) y confirmar o volver atrás. Cualquiera de las dos opciones lo lleva de vuelta a la pantalla correspondiente a la [Figura 29](#) (con la información nutricional actualizada automáticamente).

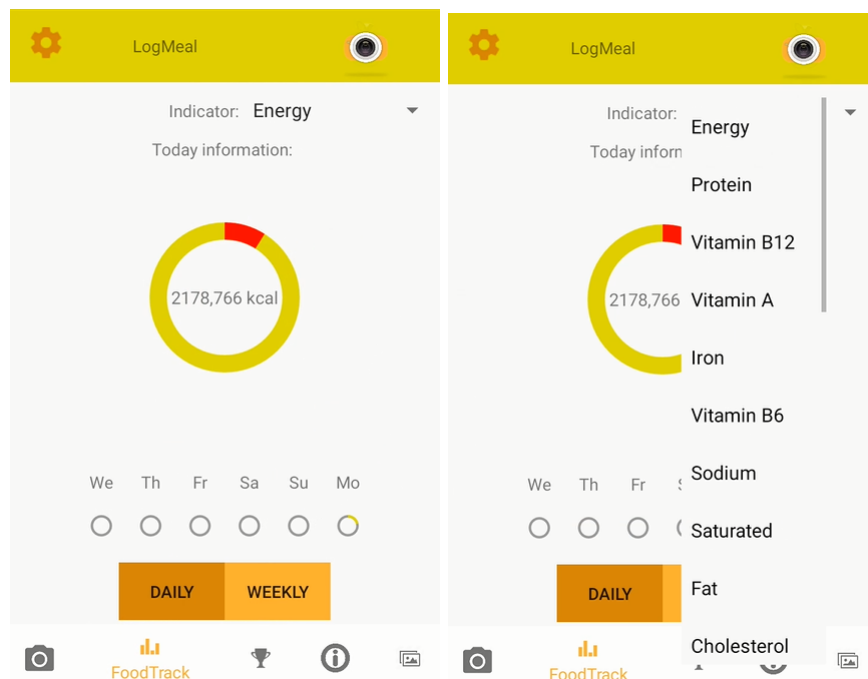


Figura 31: Pantallas de la aplicación: pantallas de estadísticas diarias organizadas por indicador nutricional

A la pantalla de la [Figura 31](#) también se accede a través del icono de la barra inferior. En estas pantallas se muestra la información de un indicador nutricional de toda la semana en detalle. Como se ve, se pueden tener datos sobre cualquier indicador que se desee. Los diagramas en forma circular se van rellenando según la proporción con respecto a la cantidad recomendada para cada indicador. El color rojo significa que está en la segunda vuelta (es decir, ya se ha pasado el límite). Si el usuario clicca sobre el botón *Weekly*, se mostrará la pantalla de la [Figura 32](#).

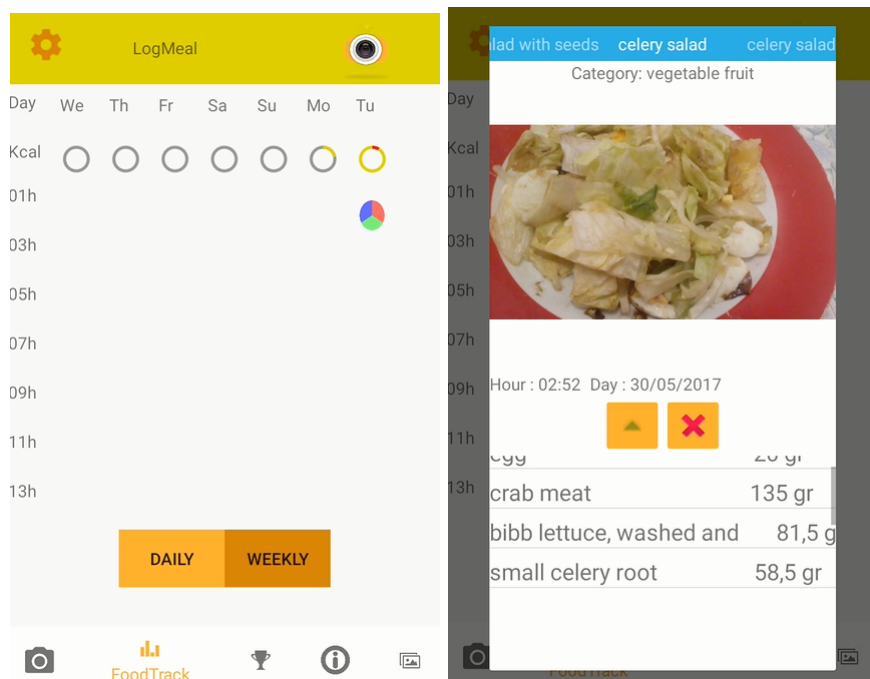


Figura 32: Pantallas de la aplicación: pantalla del diario semanal y pantalla de detalle de una de las comidas

En la pantalla de la izquierda de la [Figura 32](#) se muestra el diario de comidas, es decir todas las comidas representadas en forma de círculos organizadas por día y hora. La cantidad de colores del círculo representa la cantidad de comidas llevadas a cabo en la franja horario que representa. Si se clicca sobre cualquier círculo, se puede ver el detalle de las comidas. Por ejemplo en la imagen de la derecha se ve el resultado, en el que para cada comida se ve la imagen, la hora exacta, y la receta.



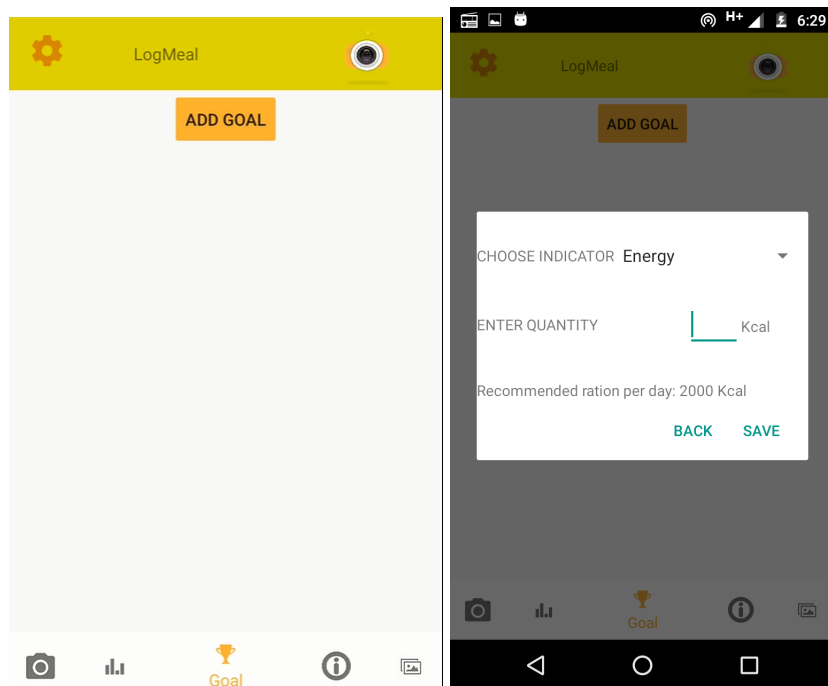


Figura 33: Pantallas de la aplicación: pantalla principal de objetivos y pantalla de añadir objetivo

A la pantalla de la [Figura 33](#) se puede acceder desde cualquier pantalla gracias al icono de la barra inferior. Vemos que no hay ningún objetivo mostrado inicialmente. El usuario puede añadir un objetivo, se mostrará la pantalla de la derecha. En esta pantalla el usuario elige el indicador nutricional sobre el que quiere crear el objetivo, y elige a continuación el valor del objetivo. Según el indicador nutricional que elija, se actualiza la información correspondiente a la cantidad recomendada por día. Ya sea confirmando o cancelando, el usuario vuelve a la pantalla general de objetivos. Si ha creado un nuevo objetivo, este se verá en la pantalla, y el usuario tendrá la posibilidad de borrarlo en cualquier momento.

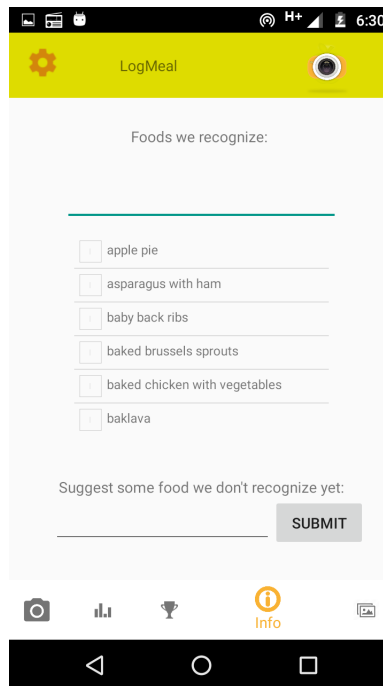


Figura 34: Pantalla de la aplicación: pantalla de información de las comidas que reconoce EatLog

A la pantalla de la [Figura 34](#) se puede acceder desde cualquier pantalla gracias al icono de la barra inferior. En esta pantalla el usuario puede consultar los platos que reconoce EatLog. También tiene la opción de sugerir un nuevo plato a reconocer.

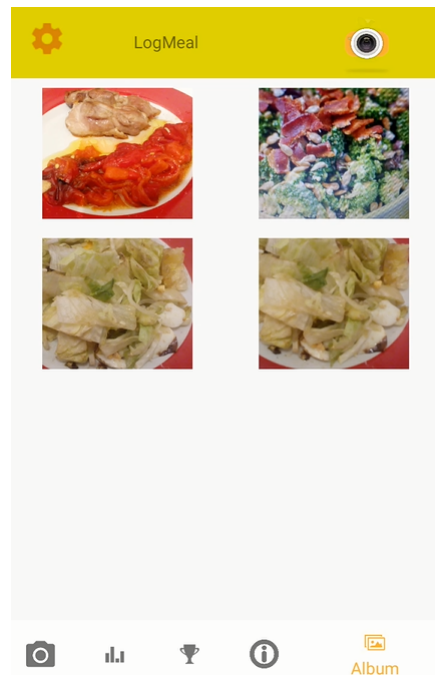


Figura 35: Pantalla de la aplicación: pantalla de álbum de imágenes tomadas a través de EatLog

A la pantalla de la [Figura 35](#) se puede acceder desde cualquier pantalla gracias al icono de la barra inferior. En esta pantalla se muestran todas las imágenes tomadas a través de EatLog.

## 5. Conclusiones y trabajos futuros

### 5.1. Conclusiones

En este proyecto, hemos explorado el problema de la construcción automática de diarios de comida. Se listarán todos los trabajos llevados a cabo:

- **Aplicación Android.** Se ha diseñado e implementado una aplicación para dispositivos móviles Android, teniendo como principales funcionalidades el registro de una comida a partir de una imagen gracias al reconocimiento automático, la traducción de la imagen a información nutricional, y la creación automática del diario de comidas con información nutricional detallada.
- **Reconocimiento de comida integrado en la aplicación.** Se han desarrollado algoritmos de reconocimiento de comida basados en la tecnología de *Deep Learning*, más concretamente en las redes neuronales convolucionales. Entre estos reconocimientos, contamos con reconocer a través de una imagen la comida (reconociendo actualmente entre 200 comidas), la categoría, los ingredientes y si la imagen corresponde a comida. Hemos conseguido integrar estos reconocimientos en la aplicación a través de la comunicación con un servidor, donde se implementaron dichos algoritmos.
- **Creación de *Datasets*.** Se ha creado una base de datos de imágenes de 99 platos de comida, con una media de 800 imágenes por comida. Por otro lado, se ha descargado información de las recetas de 189 platos (entre 10 y 300 recetas por plato), e información nutricional de 900 ingredientes (valores de 26 indicadores nutricionales).

Personalmente, este proyecto me ha contribuido a aprender en varios ámbitos. Primero de todo, he sido consciente del desarrollo de un producto desde su comienzo, analizando los requisitos, funcionalidades en continuas reuniones. Desde un principio el proyecto se integró en el marco de un equipo, presentándonos y siendo seleccionados para participar en el evento Health4Good de DKV en el mes de Noviembre de 2016. Presentamos nuestro proyecto con un prototipo muy avanzado de la aplicación, quedando en segundo lugar. Toda esta experiencia me ha hecho ver cómo trabajar en común y formar parte de un conjunto de trabajo. En términos más específicos, he obtenido grandes conocimientos en el desarrollo de aplicaciones Android y en el concepto de navegación automática. Por otro lado, me he adentrado en el mundo de los algoritmos de *Deep Learning* y más concretamente redes neuronales, entendiendo los conceptos principales gracias a haber compartido equipo con gente experta en ello. El proyecto me ha permitido aplicar todos los conocimientos obtenidos durante estos años en un contexto más orientado a la investigación y la posible aplicación en el mercado.

### 5.2. Trabajos futuros

Por un lado se mejorarán ciertos aspectos de usabilidad de la aplicación. Se mejorará la elección de la receta propuesta. Según vayan mejorando los resultados de reconocimiento de ingredientes se podrá utilizar esta información para decidir la receta que más se acerque a la imagen realizada. El desarrollo de la aplicación ha permitido mostrar de un modo sencillo el reconocimiento de imagen que aplicamos, y las posibles aplicaciones. Hemos recibido varios contactos de empresas interesadas en incluir en sus sistemas el reconocimiento que llevamos a cabo. El lanzamiento al mercado de la

aplicación ha pasado a ser menos prioritario, siendo conscientes del trabajo que aún queda para ello, y que actualmente hay opciones de mercado más directas en otro sentido. Las prioridades futuras se centrarán por un lado en la mejora del rendimiento del servidor, siendo capaces de gestionar un gran número de peticiones simultáneas. Por otro lado, nos centraremos en seguir mejorando los algoritmos de reconocimiento. Para ello estamos creando nuevas listas de platos para descargar imágenes, e investigando la descarga de otros portales (como pueden ser *Pinterest*, *Flickr*, etc). Por otro lado, hemos tenido contacto con empresas de catering que están interesadas en aplicar el reconocimiento de imagen en los comedores que gestionan. En este aspecto, deberemos adaptar y combinar los reconocimientos de imagen para poder diferenciar y reconocer diversos platos y alimentos en una única imagen.

## 6. Bibliografía

- [1] Felipe Lobelo, Phil Trotter, Ashley John Heather, Healthcare’s Rising-Risk, White paper, June 2016.
- [2] Jeffery, R. W., Epstein, L. H., Wilson, G. T., Drewnowski, A., Stunkard, A. J., & Wing, R. R., “Long-term maintenance of weight loss: current status”, *Health psychology*, 19(1S), p. 5-16, 2000
- [3] Burke, L. E., Wang, J. & Sevick , M. A., “Self-monitoring in weight loss: A systematic review of the literature”, *Journal of the American Dietetic Association*, 111, 92-102, 2011.
- [4] <https://es.wikipedia.org/wiki/Android#Arquitectura>
- [5] <https://developer.android.com/about/dashboards/index.html#platform>
- [6] <http://es.kantar.com/tech/m%C3%B3vil/2017/abril-2017-cuota-de-mercado-de-smartphones-en-espa%C3%B1a/>
- [7] <https://www.kantarworldpanel.com/es>
- [8] <https://github.com/google/volley>
- [9] Lukas Bossard, Matthieu Guillaumin, and Luc Van Gool. Food-101 – mining discriminative components with random forests. *In European Conference on Computer Vision*, 2014.
- [10] P. Herruzo and M. Bolaños and P. Radeva. Can a CNN recognize Catalan diet? *In AIP Conference Proceedings*, 2016
- [11] <http://www.seleniumhq.org/>
- [12] [https://en.wikipedia.org/wiki/Lists\\_of\\_prepared\\_foods](https://en.wikipedia.org/wiki/Lists_of_prepared_foods)
- [13] <https://developer.edamam.com/>
- [14] Marc Bolaños and Petia Radeva. Simultaneous Food Localization and Recognition. In *Computer Research Repository*, 2016.
- [15] Ashutosh Singla, Lin Yuan, and Touradj Ebrahimi. 2016. Food/Non-food Image Classification and Food Categorization using Pre-Trained GoogLeNet Model. In *Proceedings of the 2nd International Workshop on Multimedia Assisted Dietary Management (MADiMa ’16)*. ACM, New York, NY, USA, 3-11. DOI: <https://doi.org/10.1145/2986035.2986039>

## Apéndice

### Casos de usos detallados

<b>Caso de Uso 1</b>	<b>Registrarse</b>
<b>Actor Primario:</b>	Usuario
<b>Requisitos asociados:</b>	<b>RF12</b> La aplicación debe permitir al usuario identificarse de manera unívoca.
<b>Descripción:</b>	El usuario se registra eligiendo un nombre de usuario.
<b>Precondiciones:</b>	<ul style="list-style-type: none"><li>■ El usuario debe haber descargado la aplicación</li><li>■ Es la primera vez que el usuario ejecuta la aplicación</li></ul>
<b>Secuencia normal:</b> <ol style="list-style-type: none"><li>1. El usuario abre la aplicación</li><li>2. El sistema muestra la pantalla de registro</li><li>3. El usuario rellena el campo nombre de usuario</li><li>4. El usuario clicla en el botón registrarse</li><li>5. El sistema comprueba la validez del nombre de usuario</li><li>6. El sistema almacena la información del nuevo usuario creado</li><li>7. El sistema muestra la pantalla inicial</li></ol>	
<b>Postcondiciones:</b>	El usuario está registrado en EatLog.
<b>Excepciones/Extensiones:</b> <ol style="list-style-type: none"><li>5.a El nombre de usuario ya existe:<ol style="list-style-type: none"><li>1. El sistema informa que el nombre de usuario entrado ya existe.</li><li>2. El usuario vuelve al paso 3</li></ol></li></ol>	
<b>Frecuencia:</b>	Baja
<b>Importancia:</b>	Alta

***Comentarios:***

Esta opción solo es mostrada una vez. Una vez el usuario se registra, ese nombre de usuario queda asociado a su terminal móvil.

---

Tabla 12: Caso de Uso 1. Registrarse



<b>Caso de Uso 2</b>	<b>Hacer Foto</b>
<b>Actor Primario:</b>	Usuario
<b>Requisitos asociados:</b>	<b>RF1</b> La aplicación debe permitir al usuario realizar una foto.
<b>Descripción:</b>	El usuario realiza una foto con su terminal móvil.
<b>Precondiciones:</b>	<ul style="list-style-type: none"> <li>■ El usuario está registrado</li> <li>■ El usuario tiene instalada una aplicación de captura de imágenes en su terminal móvil</li> </ul>
<b>Secuencia normal:</b>	
<ol style="list-style-type: none"> <li>1. El usuario abre la aplicación</li> <li>2. El sistema muestra la pantalla principal</li> <li>3. El usuario elige la opción de realizar una foto ('FoodSnap')</li> <li>4. El sistema conecta con la aplicación de captura de imágenes</li> <li>5. El usuario realiza la foto</li> <li>6. El usuario confirma la foto realizada</li> <li>7. La imagen es enviada al servidor</li> </ol>	
<b>Postcondiciones:</b>	La imagen queda registrada en EatLog y se guarda en el servidor.
<b>Excepciones/Extensiones:</b>	
6.a El usuario descarta la foto realizada: <ol style="list-style-type: none"> <li>1. El usuario vuelve al paso 5</li> </ol>	
<b>Frecuencia:</b>	Alta
<b>Importancia:</b>	Alta
<b>Comentarios:</b>	En un principio se implementó la captura de imagen en EatLog, manteniendo el diseño y estructura. Los problemas de compatibilidades con las diferentes versiones de cámaras (flash, autoenfoco, etc) provocaron esta externalización de la captura de imagen.

<b>Caso de Uso 3</b>	<b>Subir Foto</b>
<b>Actor Primario:</b>	Usuario
<b>Requisitos asociados:</b>	<b>RF4</b> La aplicación debe permitir al usuario subir a EatLog una imagen contenida en su terminal.
<b>Descripción:</b>	El usuario sube una imagen contenida en su terminal móvil a EatLog.
<b>Precondiciones:</b>	<ul style="list-style-type: none"> <li>■ El usuario está registrado</li> <li>■ El usuario tiene instalada una aplicación que muestre la galería de imágenes en su terminal móvil</li> </ul>
<b>Secuencia normal:</b>	
<ol style="list-style-type: none"> <li>1. El usuario abre la aplicación</li> <li>2. El sistema muestra la pantalla principal</li> <li>3. El usuario elige la opción de subir una comida manualmente ('Manual Log')</li> <li>4. El sistema muestra la pantalla de opciones de subida manual</li> <li>5. El usuario elige la opción de subir una imagen existente ('Upload Picture')</li> <li>6. El sistema conecta con la aplicación de galería de imágenes</li> <li>7. El usuario elige una imagen</li> <li>8. La imagen es enviada al servidor</li> </ol>	
<b>Postcondiciones:</b>	La imagen queda registrada en EatLog y se guarda en el Servidor
<b>Excepciones/Extensiones:</b>	
<b>Frecuencia:</b>	Alta
<b>Importancia:</b>	Alta
<b>Comentarios:</b>	

Tabla 16: Caso de Uso 3. Subir Foto

<b>Caso de Uso 4</b>	<b>Subir Comida</b>
<b>Actor Primario:</b>	Usuario
<b>Requisitos asociados:</b>	<b>RF100</b> La aplicación debe permitir al usuario subir una comida sin imagen.
<b>Descripción:</b>	El usuario sube en EatLog una comida sin imagen.
<b>Precondiciones:</b>	<ul style="list-style-type: none"> <li>▪ El usuario está registrado</li> </ul>
<b>Secuencia normal:</b>	
<ol style="list-style-type: none"> <li>1. El usuario abre la aplicación</li> <li>2. El sistema muestra la pantalla principal</li> <li>3. El usuario elige la opción de subir una comida manualmente ('Manual Log')</li> <li>4. El sistema muestra la pantalla de opciones de subida manual</li> <li>5. El usuario elige la opción de registrar una comida de forma manual ('Manual Input')</li> <li>6. El sistema muestra una pantalla con un campo de texto y una lista de comidas</li> <li>7. El usuario elige o escribe la comida</li> <li>8. El usuario confirma la opción escogida</li> <li>9. El sistema muestra una pantalla con un reloj y un calendario</li> <li>10. El usuario escoge hora y fecha</li> </ol>	
<b>Postcondiciones:</b>	
<b>Excepciones/Extensiones:</b>	
<b>Frecuencia:</b>	Baja
<b>Importancia:</b>	Alta
<b>Comentarios:</b>	El sistema muestra las comidas que se reconocen, con el objetivo de priorizar la elección de una de estas ante la escritura de una nueva comida ya que el sistema tiene recetas e información sobre las comidas que reconoce.

Tabla 18: Caso de Uso 4. Subir Comida

<b>Caso de Uso 7</b>	<b>Consultar Estadísticas</b>
<b>Actor Primario:</b>	Usuario
<b>Requisitos asociados:</b>	<b>RF7</b> La aplicación debe permitir al usuario consultar información de cada indicador nutricional.
<b>Descripción:</b>	El usuario consulta la información diaria y semanal de cada uno de los 26 indicadores nutricionales registrados en EatLog.
<b>Precondiciones:</b>	
<b>Secuencia normal:</b>	<ol style="list-style-type: none"> <li>1. El usuario clics sobre la sección de la barra inferior correspondiente a Estadísticas ('FoodTrack')</li> <li>2. El sistema muestra una pantalla con la información de un indicador nutricional por defecto</li> <li>3. El usuario clics sobre el desplegable de indicadores nutricionales</li> <li>4. El sistema muestra todos los indicadores nutricionales que gestiona EatLog</li> <li>5. El usuario elige uno de los indicadores</li> <li>6. El sistema muestra en pantalla diferentes gráficas con la información semanal y de los últimos 7 días por separado del indicador nutricional correspondiente</li> </ol>
<b>Postcondiciones:</b>	
<b>Excepciones/Extensiones:</b>	
<b>Frecuencia:</b>	Media
<b>Importancia:</b>	Media
<b>Comentarios:</b>	En todos los indicadores la estructura de gráficas es la misma, salvo para las calorías. Para la energía consumida (en Kcal) es importante la información diaria. No obstante, para el resto de indicadores (vitaminas, minerales, etc) la información de la media de los últimos 7 días es lo más importante.

Tabla 20: Caso de Uso 7. Consultar Estadísticas

<b>Caso de Uso 8</b>	<b>Consultar comidas</b>
<b>Actor Primario:</b>	Usuario
<b>Requisitos asociados:</b>	<b>RF7</b> La aplicación debe permitir al usuario consultar su diario semanal.
<b>Descripción:</b>	El usuario consulta todas las comidas de la última semana en un formato organizado por días y horas.
<b>Precondiciones:</b>	
<b>Secuencia normal:</b>	<ol style="list-style-type: none"> <li>1. El usuario clics sobre la sección de la barra inferior correspondiente a Estadísticas ('FoodTrack')</li> <li>2. El sistema muestra una pantalla con las estadísticas de los indicadores nutricionales</li> <li>3. El usuario clics sobre el botón correspondiente a la información semanal, 'Weekly'</li> <li>4. El sistema muestra una pantalla con un gráfico semanal de las comidas (representadas por círculos) organizadas por día y hora.</li> <li>5. El usuario clics sobre uno de los círculos correspondiente a una franja horaria en la que haya registrada una o varias comidas.</li> <li>6. El sistema muestra una pantalla con la información de la/s comida/s del día y franja horaria seleccionados.</li> </ol>
<b>Postcondiciones:</b>	
<b>Excepciones/Extensiones:</b>	<ol style="list-style-type: none"> <li>6.a El sistema da la opción de borrar una de las comidas registradas que se muestran: <ol style="list-style-type: none"> <li>1. Correspondiente a Caso de Uso 9</li> </ol> </li> </ol>
<b>Frecuencia:</b>	Media
<b>Importancia:</b>	Alta
<b>Comentarios:</b>	

Tabla 22: Caso de Uso 8. Consultar Comidas

<b>Caso de Uso 9</b>	<b>Borrar Comida</b>
<b>Actor Primario:</b>	Usuario
<b>Requisitos asociados:</b>	<b>RF7</b> La aplicación debe permitir al usuario borrar una comida de su diario de comidas.
<b>Descripción:</b>	El usuario elimina una comida que ha registrado con anterioridad en EatLog.
<b>Precondiciones:</b>	<ul style="list-style-type: none"> <li>El usuario está en la Extensión 6.a del Caso de Uso 8, es decir, el usuario ha clicado en una franja horaria con alguna comida registrada.</li> </ul>
<b>Secuencia normal:</b>	
<ol style="list-style-type: none"> <li>1. El sistema muestra una pantalla con información sobre la comida (imagen, ingredientes)</li> <li>2. El usuario clicla sobre el botón de borrar comida (señalizado con una 'X')</li> <li>3. El sistema muestra una ventana emergente avisando al usuario de la acción que va a llevar a cabo</li> <li>4. El usuario confirma el borrado de la comida</li> <li>5. El sistema actualiza el almacenamiento interno de las comidas</li> <li>6. El sistema actualiza la gráfica del diario semanal</li> </ol>	
<b>Postcondiciones:</b>	<ul style="list-style-type: none"> <li>La comida elegida es borrada de la Base de Datos del terminal móvil</li> </ul>
<b>Excepciones/Extensiones:</b>	
4.a El usuario no confirma el borrado de la comida: <ol style="list-style-type: none"> <li>1. El usuario vuelve al paso 1</li> </ol>	
<b>Frecuencia:</b>	Baja
<b>Importancia:</b>	Media
<b>Comentarios:</b>	

Tabla 24: Caso de Uso 9. Borrar Comida

Caso de Uso 10	Consultar Platos Reconocibles
<b>Actor Primario:</b>	Usuario
<b>Requisitos asociados:</b>	<b>RF9</b> La aplicación debe permitir al usuario consultar información sobre las comidas que reconoce el sistema.
<b>Descripción:</b>	El usuario consulta la lista de platos que reconoce el sistema.
<b>Precondiciones:</b>	
<b>Secuencia normal:</b>	<ol style="list-style-type: none"> <li>1. El usuario clicla sobre la sección de la barra inferior correspondiente a Información</li> <li>2. El sistema muestra una pantalla con la lista de platos que reconoce EatLog, un campo de texto para buscar y un campo de texto para sugerir nuevos platos.</li> </ol>
<b>Postcondiciones:</b>	
<b>Excepciones/Extensiones:</b>	<ol style="list-style-type: none"> <li>2.a El usuario busca si el sistema reconoce un plato: <ol style="list-style-type: none"> <li>1. El usuario escribe en el campo de texto correspondiente a buscar platos.</li> <li>2. El sistema actualiza el autocompletado del plato según lo que escriba el usuario</li> </ol> </li> <li>2.b El usuario propone un nuevo plato: <ol style="list-style-type: none"> <li>1. Correspondiente a Caso de Uso 11</li> </ol> </li> </ol>
<b>Frecuencia:</b>	Baja
<b>Importancia:</b>	Media
<b>Comentarios:</b>	

Tabla 26: Caso de Uso 10. Consultar Platos Reconocibles



Caso de Uso 11	Proponer Nuevo Plato
<i>Actor Primario:</i>	Usuario
<i>Requisitos asociados:</i>	<b>RF13</b> El sistema debe ser capaz de mejorar sus algoritmos de reconocimiento gracias a información obtenida a través de EatLog
<i>Descripción:</i>	El usuario propone algún plato que el sistema de EatLog aún no reconoce.
<i>Precondiciones:</i>	<ul style="list-style-type: none"> <li>■ El usuario está en la Extensión 2.b del Caso de Uso 10, es decir, ha clicado en la sección de la barra inferior correspondiente a Información</li> </ul>
<i>Secuencia normal:</i>	
<ol style="list-style-type: none"> <li>1. El usuario escribe sobre el campo de texto habilitado el nombre del plato que quiere proponer/sugerir</li> <li>2. El usuario clicla sobre el botón que confirma la proposición/sugerencia</li> <li>3. El sistema procesa la petición y almacena la información</li> </ol>	
<i>Postcondiciones:</i>	<ul style="list-style-type: none"> <li>■ La petición está almacenada en el Servidor</li> </ul>
<i>Excepciones/Extensiones:</i>	
<i>Frecuencia:</i>	Baja
<i>Importancia:</i>	Media
<i>Comentarios:</i>	

Tabla 28: Caso de Uso 11. Proponer Nuevo Plato

Caso de Uso 12	Añadir Objetivos
<b>Actor Primario:</b>	Usuario
<b>Requisitos asociados:</b>	<b>RF8</b> La aplicación debe permitir al usuario gestionar sus objetivos sobre indicadores nutricionales.
<b>Descripción:</b>	El usuario gestiona sus objetivos (añadir y borrar) con respecto al valor diario ingestado de cualquiera de los 26 indicadores nutricionales que gestiona EatLog.
<b>Precondiciones:</b>	
<b>Secuencia normal:</b>	<ol style="list-style-type: none"> <li>1. El usuario clicca sobre la sección de la barra inferior correspondiente a Objetivos ('Goal')</li> <li>2. El sistema muestra una pantalla con la lista de los objetivos actuales y un botón que permite añadir objetivos</li> <li>3. El usuario clicca sobre el botón de añadir objetivo</li> <li>4. El sistema muestra una ventana emergente con un desplegable correspondiente a los indicadores nutricionales y un campo de texto para escribir el valor del objetivo</li> <li>5. El usuario elige entre los indicadores</li> <li>6. El sistema actualiza la medida del objetivo que marcará el usuario (kcal, mg, g, etc)</li> <li>7. El sistema actualiza la información mostrada acerca de la ingesta recomendada por día</li> <li>8. El usuario escribe el valor numérico del objetivo</li> <li>9. El usuario confirma el objetivo</li> <li>10. El sistema almacena el nuevo objetivo en la base de datos del terminal móvil</li> <li>11. Si es el único objetivo, el sistema cataloga al indicador nutricional correspondiente como indicador por defecto para las estadísticas</li> </ol>
<b>Postcondiciones:</b>	<ul style="list-style-type: none"> <li>■ El objetivo añadido está almacenado en la base de datos de EatLog</li> </ul>

***Excepciones/Extensiones:***

2.a El usuario elimina uno de los objetivos existentes:

1. El usuario clicca sobre el botón de borrar (icono de 'papelera') del objetivo que desea borrar.
2. El sistema borra el objetivo clicado del almacenamiento del terminal móvil

9.a El usuario no confirma el objetivo y sale de la ventana emergente:

1. El usuario vuelve al paso 2

<b><i>Frecuencia:</i></b>	Baja
<b><i>Importancia:</i></b>	Baja
<b><i>Comentarios:</i></b>	

Tabla 30: Caso de Uso 12. Añadir Objetivos

Caso de Uso 14	Guardar Información Sobre Imagen
<i>Actor Primario:</i>	Servidor
<i>Requisitos asociados:</i>	<b>RF11</b> El sistema debe almacenar los datos aportados por el usuario ('feedback' del reconocimiento)
<i>Descripción:</i>	El servidor almacena en una base de datos información confirmada de una imagen que ha procesado previamente.
<i>Precondiciones:</i>	<ul style="list-style-type: none"> <li>■ El servidor ha procesado la misma imagen y enviado la información del reconocimiento</li> <li>■ El usuario ha confirmado los datos sobre el reconocimiento de la imagen</li> <li>■ El servidor ha recibido la información (desde la aplicación EatLog)</li> </ul>
<i>Secuencia normal:</i>	
<ol style="list-style-type: none"> <li>1. El servidor hace una petición a la base de datos para añadir la información predicha por el reconocimiento y confirmada de la imagen</li> <li>2. La base de datos procesa la petición</li> </ol>	
<i>Postcondiciones:</i>	<ul style="list-style-type: none"> <li>■ El sistema tiene almacenada información confirmada acerca de una imagen.</li> </ul>
<i>Excepciones/Extensiones:</i>	
<i>Frecuencia:</i>	Alta
<i>Importancia:</i>	Alta
<i>Comentarios:</i>	

Tabla 32: Caso de Uso 14. Guardar Información sobre Imagen

Caso de Uso 15	Guardar Información Sobre Usuario
<i>Actor Primario:</i>	Servidor
<i>Requisitos asociados:</i>	<b>RF14</b> El sistema debe almacenar datos del usuario y personalizar sus funcionalidades.
<i>Descripción:</i>	El servidor almacena en una base de datos información acerca de las comidas y recetas del usuario.
<i>Precondiciones:</i>	<ul style="list-style-type: none"> <li>■ El usuario ha confirmado el registro de una comida</li> <li>■ El servidor ha recibido la información (desde la aplicación EatLog)</li> </ul>
<i>Secuencia normal:</i>	
<ol style="list-style-type: none"> <li>1. El servidor hace una petición a la base de datos para añadir la información de la comida registrada asociada a una imagen, una receta y un usuario</li> <li>2. La base de datos procesa la petición</li> </ol>	
<i>Postcondiciones:</i>	<ul style="list-style-type: none"> <li>■ El sistema tiene almacenada información acerca de la comida y receta de un usuario.</li> </ul>
<i>Excepciones/Extensiones:</i>	
<i>Frecuencia:</i>	Alta
<i>Importancia:</i>	Alta
<i>Comentarios:</i>	

Tabla 34: Caso de Uso 15. Guardar Información sobre usuario